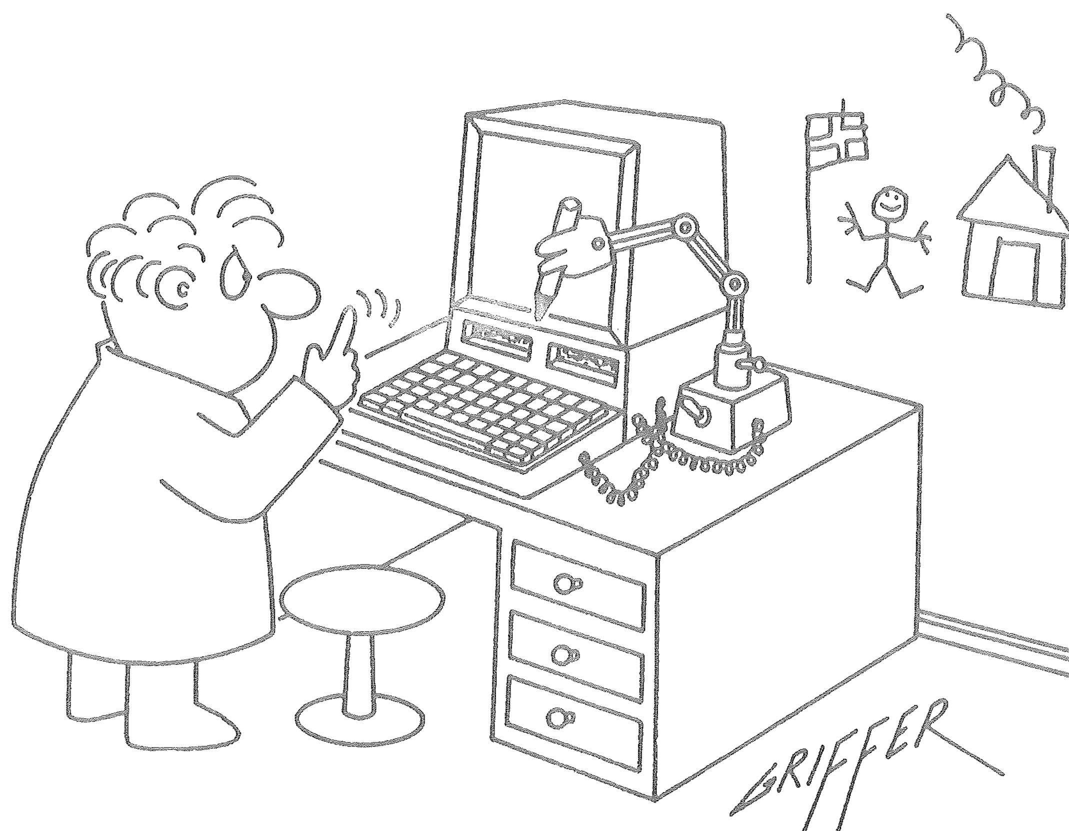
SYNTAX

SVI · MSX · PC

ÅRGANG 6 NR. 2 PRIS KR. 17



MARTS 1989

UDGIVET AF: MSX BRUGER KLUBBEN

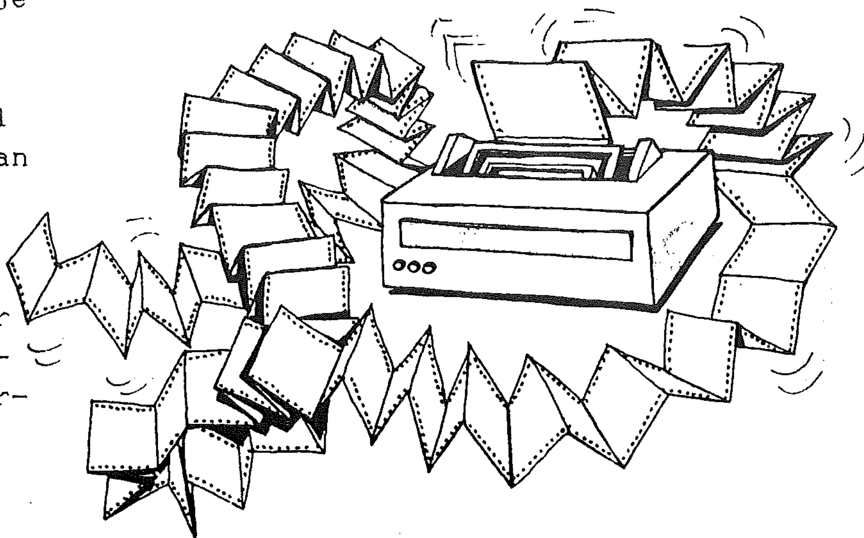
INDHOLDSFORTEGNELSE PÅ BÅND.

Der må da være andre end mig der ikke bruger disketter, men gemmer programmer på bånd? Selvfølgelig noterer du altid ned hvad der er på dine bånd, men måske kender du en der har fundet nogle gamle bånd med ukendt indhold? Når man skriver CLOAD får man det første program som er gemt med CSAVE, men måske overser man et program som er gemt med SAVE eller en datafil. Eller noget som er gemt med BSAVE. Dette program viser navnene på alle filer af de tre typer på et bånd. Det er nemmest at få navnefortegnelsen på tryk ved at skrive LPRINT USR(0). Så kan du drikke aftente mens båndet kører og indholdsfortegnelsen bliver skrevet.

Vi synes alle at det tager urimeligt lang tid at lagre data på bånd og hente dem igen. Officielt skal man ikke bruge 2400 baud med en almindelg kassettebåndoptager, men det virker jo alligevel! Jeg har haft gode resultater med at bruge krombånd på en billig kassettebåndoptager, som ikke er beregnet til krombånd. Netop derved fremhæves de høje frekvenser.

Ved programmer som gemmes med SAVE og ved datafiler bruges en stor del af tiden til at skrive headere foran dataklumper på 256 bytes. Med 2400 baud varer en enkelt header 1,65 sek., medens det tager 1,2 sek. at skrive de 256 bytes. Uden problemer har jeg kunnet nedsætte headerlængden til næsten en trediedel. Headerlængden for 2400 baud findes på adresse &HF405, og den er normalt 31. Prøv først med POKE &HF405,11 efterfulgt af SCREEN 0,0,0,2. Det kan nok spare kostbare minutter af din restlevetid !

```
10 'Dette program viser alle filer
20 'på et bånd af typerne CLOAD,
30 'BLOAD og LOAD.
40 'Det kan placeres hvorsomhelst i
50 'hukommelsen, bare der er plads
60 'til de 142 bytes.
70 'Skriv PRINT USR(0) for at se
80 'filoversigten. Virker også med
90 'LPRINT.
100 DEFINT A-Z
110 START = &HE000 ' Kan ændres !
120 DEFUSR = START
130 FOR AD = START TO START + 141
140 READ N: POKE AD,N
150 NEXT
1000 DATA 243,33,225,233,34,246,247
1010 DATA 205,246,247,17,22,0,25,84
1020 DATA 93,1,82,0,9,235,115,35,114
1030 DATA 205,225,0,56,64,6,16,33,114
1040 DATA 224,197,229,205,228,0,225
1050 DATA 193,56,50,119,35,16,243,17
1060 DATA 240,255,25,6,9,126,35,190
1070 DATA 32,222,16,250,17,11,0,25,1
1080 DATA 67,66,112,254,208,40,12,113
1090 DATA 254,211,40,7,6,32,112,254
1100 DATA 234,32,196,17,246,255,25
1110 DATA 205,120,102,24,187,205,183
1120 DATA 0,205,231,0,62,3,50,99,246
1130 DATA 33,11,0,34,248,247,216,195
1140 DATA 178,115,0,0,0,0,0,0,0,0,0
1150 DATA 0,0,0,0,0,0,32,32,9,9,32,76
1160 DATA 79,65,68,13,10,0
```



INDHOLDFORTEGNELSE

Artikler, annoncer eller programmer der ønskes optaget i bladet, sendes til Kim Andersen eller Preben Lund Bånd og disk sendes retur.

LEDER

Indtil videre har det stået noget sløjd til med indbetalingerne til den nye årgang af Syntax'en. Dette selvfølgelig meget bedrøveligt, da hele vores budget bæres oppe af medlems-kontingentet. Imidlertid håber vi på at medlemsantallet vil rette sig op i den næste månedstid.

De fleste af dem, der har indbetalt, har på girokortet skrevet om deres mening, ønsker osv. angående bladet. Alt i alt er det heldigvis et meget positivt billede, der tegner sig. De punkter vi har fået ris ved, vil vi naturligvis tage op til revision, og dem vi har fået ros ved, vil vi forstærke efter bedste evne. De mange ønsker vi har fået er kommet i hænderne på alle redaktionsmedlemmerne, så vi kan forme de kommende numre af Syntax'en til størst mulig glæde for læserne.

Kim Andersen

O P L Y S N I N G E R

Foreningen hedder: MSX BRUGERKLUBBEN
Medlemsskab koster 165 Kr. (årligt)
Salgsannoncer er for medlemmer gratis
Annoncører : 1/2=300 kr. 1/1=500 kr.

Medlemskab opnås ved henvendelse til foreningens kasserer, eller ved indbetaling af kontingent på postgironr.
8 2 0 6 0 8 2
HUSK at opgive navn, adresse, postnr. og by, samt tlf.nr. og maskintype.

Bladet hedder : SYNTAX
Oplag : 800 eks.
Udkommer : 10 gange årligt (ikke jan og juli).
Tryk : Foto offset.
Løssalgspris : 17,00 Kr.
Udkommer næste gang : April

Formand :	Kim Andersen
	Fundervej 32
	2610 Rødovre
Tlf. nr.	02 94 26 74
Kasserer :	Preben Lund
	Galgebakken Neder 2-3 b.
	2620 Albertslund
Tlf. nr.	02 64 76 26

Redaktionsgruppen består af formand, kasser og følgende :

Richard Foersom	Tlf.nr.	01 22 26 34
Henrik Larsen	-	01 74 73 99
Peter Knudsen	-	02 17 76 23
Søren Mortensen	-	01 69 77 40
Jannik Storm	-	01 65 74 59
Per Underlien	-	02 64 09 66
Henrik Gilvad	-	03 14 36 57
Søren Nielsen	-	02 52 85 93
Jonas Tromborg	-	02 65 28 98
Lars Hansen	-	02 32 44 73
FYN		
Jens Møller	-	09 94 22 75
JYLLAND		
Johnny B.	-	06 82 66 98
SVERIGE		
Jan Bojstrup	00946.	40-22 23 33
Joakim Tornhill	.	40-91 96 58
Magnus Balldin	.	40-15 01 98
NORGE		
Kjell Johansen	00947.	02-28 24 45

PLAY DIREKTE I DATA LINIER.

HEJ FANS

Hermed et nytteprogram til Jer der sidder og laver spil og lignende. Tit mangler man lige en stump musik til at sætte prikken over i'et og dette kan du klare med mit lille program fra Japan.

Jeg har bearbejdet det lidt for at man skal kunne forstå det og jeg har også tilføjet lidt extra bl. a. en pause funktion, er du lidt opfindsom kan du sagtens selv bygge mere på programmet som her kun bruger 1/4 dels noder.

Du får Taster på skærmen og kan så vælge mellem orgel eller klaver lyd Derefter kan du spille melodien på computer tasterne, husk pauser, når du er færdig tryk return og ind giv filnavn, derefter bliver melodien gemt ud i en datafil som Aschi det vil sige at filen kan Merges med dit eget Basic program. Linie numrene med Play starter ved 30000 i datafilen.

Du kan også loade filen og lade din computer spille stykket.

Computeren spiller også melodien mens den gemmer den ud på dit Data medie (Bånd eller Disk)

Jeg håber at nogen kan få fornøjelse af dette program, hvis du nu forbedrer det så send resultatet til mig, så vi alle kan være med og drage nytte deraf.

Obs medlemmer der sender noget til mig får altid en god belønning.

Efter programmet en lille prøve på hvad man kan gøre, det har jeg selv indspillet.

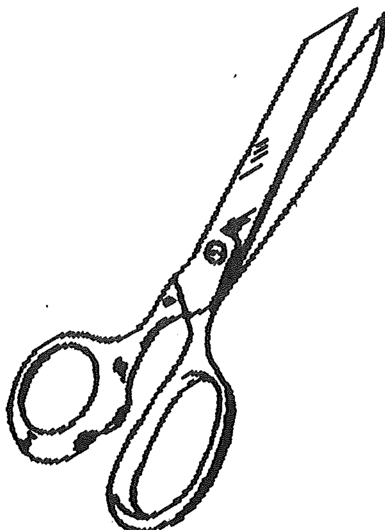
Hej fra Johnny B Silkeborg Danmark

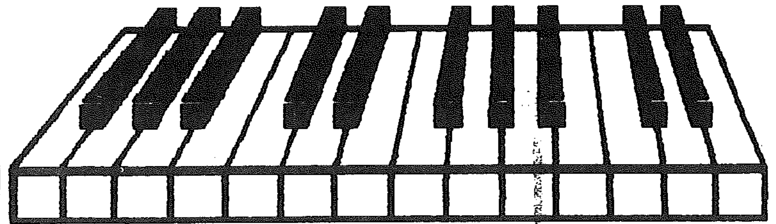
```
10 '****************************
20 '*         BASIC PLAY       *
30 '* Bearbejdet af Johnny B   *
40 '*       org. prg. Japan    *
50 '* Spil på tasterne og få   *
60 '* melodien lagt i en data  *
70 '* fil (Bånd & Disk)        *
80 '*        SYNTAXEN 89       *
90 '****************************
95 '
100 SCREEN 1
110 GOSUB 1340
120 CLEAR:GOSUB 300
130 DIM KY(1000):P=0:SW=0:N=30000
140 GOSUB 930
150 LOCATE 2,4:INPUT"(1) Klaver  (2
) Orgel ";TY$
160 LOCATE 2,4:PRINT"
                     "
170 IF TY$="1" THEN LOCATE 4,4:PRIN
T"Klaverlyd "
180 IF TY$="2" THEN LOCATE 4,4:PRIN
T"Orgellyd"
190 GOSUB 340
200 A$=INKEY$:IF A$="" GOTO 200
210 IF ASC(A$)=&HD GOTO 780
220 IF ASC(A$) > &H60 THEN A$=CHR$(
ASC(A$)-&H20)
230 F=INSTR(K$,A$)
240 IF F=0 GOTO 200
250 GOSUB 280
260 IF P=1000 THEN GOTO 780
270 GOTO 200
280 ON F GOSUB 430,440,450,460,470,
480,490,500,510,520,530,540,550,560
,570,580,590,600
290 RETURN
300 '
310 READ A$
320 IF A$<>"" THEN K$=K$+A$:GOTO 31
0
330 RETURN
340 IF TY$="2" THEN 390
350 MV$="M4000"
360 LV$="L8"
370 SV$="S0"
380 RETURN
390 MV$="V13"
400 LV$="L8"
410 SV$=""
420 RETURN
430 X$="O4C" :GOTO 610:' "Z"
440 X$="O4D" :GOTO 610:' "X"
450 X$="O4E" :GOTO 610:' "C"
460 X$="O4F" :GOTO 610:' "V"
```

```
470 X$="O4G" :GOTO 610:' "B"
480 X$="O4A" :GOTO 610:' "N"
490 X$="O4B" :GOTO 610:' "M"
500 X$="O5C" :GOTO 610:' ","
510 X$="O4C#":GOTO 610:' "S"
520 X$="O4D#":GOTO 610:' "D"
530 X$="O4F#":GOTO 610:' "G"
540 X$="O4G#":GOTO 610:' "G"
550 X$="O4A#":GOTO 610:' "J"
560 X$="O5C#":GOTO 610:' "L"
570 X$="R":GOTO 610:' "P"
580 RETURN
590 RETURN
600 RETURN
610 '
620 ' Spille
630 '
640 Y$=LV$+SV$+MV$+X$
650 PLAY Y$
660 IF SW=1 GOTO 700
670 KY(P)=F:P=P+1
680 LOCATE 2,11:PRINT "NR";1000-P
690 RETURN
700 PRINT N;" PLAY";CHR$(&H22);X$;C
HR$(&H22)
710 PRINT #1,N;" PLAY";CHR$(&H22);X
$;CHR$(&H22)
720 RETURN
730 '
740 ' Hvilke taster bruges
750 '
760 DATA "Z","X","C","V","B","N","M
",","
770 DATA "S","D","G","H","J","L","P
","
780 '
790 ' Gemme det spillede som Aschi
fil
800 '
810 CLS:INPUT"FILNAVN";FI$
820 OPEN FI$+".OTO" FOR OUTPUT AS #
1
830 PRINT N;" PLAY";CHR$(&H22);LV$;
SV$;MV$;CHR$(&H22):N=N+10
840 PRINT #1,N;" PLAY";CHR$(&H22);L
V$;SV$;MV$;CHR$(&H22):N=N+10
850 I=0:SW=1
860 IF KY(I) = 0  GOTO 890
870 F =KY(I):GOSUB 280
880 N=N+10:I=I+1:GOTO 860
890 CLOSE #1
900 INPUT"RETRY";S$
910 IF S$="Y" OR S$="y" GOTO 120
920 END
930 '
940 ' Skærmbillede
950 '
960 KEY OFF:CLS
970 LOCATE 3,13:PRINTCHR$(1);CHR$(2
4+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(18+64);CHR$
(1);CHR$(23+64);CHR$(1);CHR$(23+64)
;CHR$(1);CHR$(25+64);"  ";
980 LOCATE12,13:PRINTCHR$(1);CHR$(2
4+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(18+64);CHR$
(1);CHR$(23+64);CHR$(1);CHR$(23+64)
;CHR$(1);CHR$(18+64);CHR$(1);CHR$(2
3+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(25+64);
990 LOCATE24,13:PRINTCHR$(1);CHR$(2
4+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(25+64);
1000 LOCATE 3,14:PRINTCHR$(1);CHR$(
22+64);"S ";CHR$(1);CHR$(22+64);"D
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);"G ";CHR$(1);CHR$(22+64
);"H "
1010 LOCATE18,14:PRINTCHR$(1);CHR$(
22+64);"J ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"L ";CHR$(1);
CHR$(22+64);
1020 LOCATE 3,15:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);"  ";CHR$(1);CHR$(22+64
);"  "
1030 LOCATE18,15:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);
1040 LOCATE 3,16:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);"  ";CHR$(1);CHR$(22+64
);"  "
1050 LOCATE18,16:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);
1060 LOCATE 3,17:PRINTCHR$(1);CHR$(
22+64);"C#";CHR$(1);CHR$(22+64);"D#
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);"F#";CHR$(1);CHR$(22+64
);"G#"
1070 LOCATE18,17:PRINTCHR$(1);CHR$(
22+64);"A#";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"C#";CHR$(1);
CHR$(22+64);
```

```
1080 LOCATE 2,18:PRINTCHR$(1);CHR$(
24+64);CHR$(1);CHR$(17+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(18+64);CHR
$(1);CHR$(17+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(18+64);CHR$(1);CHR$(
17+64);CHR$(1);CHR$(23+64);
1090 LOCATE11,18:PRINTCHR$(1);CHR$(
18+64);CHR$(1);CHR$(17+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(18+64);CHR
$(1);CHR$(17+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(18+64);CHR$(1);CHR$(
17+64);CHR$(1);CHR$(23+64);
1100 LOCATE20,18:PRINTCHR$(1);CHR$(
18+64);CHR$(1);CHR$(17+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(18+64);CHR
$(1);CHR$(17+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(18+64);CHR$(1);CHR$(
27+64);
1110 LOCATE 2,19:PRINTCHR$(1);CHR$(
22+64);"Z ";CHR$(1);CHR$(22+64);"X
";CHR$(1);CHR$(22+64);"C ";CHR$(1);
CHR$(22+64);"V ";CHR$(1);CHR$(22+64
);"B "
1120 LOCATE17,19:PRINTCHR$(1);CHR$(
22+64);"N ";CHR$(1);CHR$(22+64);"M
";CHR$(1);CHR$(22+64);", ";CHR$(1);
CHR$(22+64);
1130 LOCATE 2,20:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);"  ";CHR$(1);CHR$(22+64
);"  "
1140 LOCATE17,20:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);
1150 LOCATE 2,21:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);"  ";CHR$(1);CHR$(22+64
);"  "
1160 LOCATE17,21:PRINTCHR$(1);CHR$(
22+64);"  ";CHR$(1);CHR$(22+64);"
";CHR$(1);CHR$(22+64);"  ";CHR$(1);
CHR$(22+64);
1170 LOCATE 2,22:PRINTCHR$(1);CHR$(
22+64);"C ";CHR$(1);CHR$(22+64);"D
";CHR$(1);CHR$(22+64);"E ";CHR$(1);
CHR$(22+64);"F ";CHR$(1);CHR$(22+64
);"G "
1180 LOCATE17,22:PRINTCHR$(1);CHR$(
22+64);"A ";CHR$(1);CHR$(22+64);"H
";CHR$(1);CHR$(22+64);"C ";CHR$(1);
CHR$(22+64);
1190 LOCATE 2,23:PRINTCHR$(1);CHR$(
26+64);CHR$(1);CHR$(23+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(17+64);CHR
$(1);CHR$(23+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(17+64);CHR$(1);CHR$(
23+64);CHR$(1);CHR$(23+64);
1200 LOCATE11,23:PRINTCHR$(1);CHR$(
17+64);CHR$(1);CHR$(23+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(17+64);CHR
$(1);CHR$(23+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(17+64);CHR$(1);CHR$(
23+64);CHR$(1);CHR$(23+64);
1210 LOCATE20,23:PRINTCHR$(1);CHR$(
17+64);CHR$(1);CHR$(23+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(17+64);CHR
$(1);CHR$(23+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(27+64);
1220 LOCATE 19,6:PRINTCHR$(1);CHR$(
24+64);CHR$(1);CHR$(23+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(23+64);CHR
$(1);CHR$(23+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(23+64);CHR$(1);CHR$(
23+64);CHR$(1);CHR$(25+64);
1230 LOCATE 2,7:PRINTCHR$(1);CHR$(2
4+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(23+64);CHR$
(1);CHR$(23+64);CHR$(1);CHR$(23+64)
;CHR$(1);CHR$(23+64);CHR$(1);CHR$(2
3+64);
1240 LOCATE 8,7:PRINTCHR$(1);CHR$(2
3+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(23+64);CHR$
(1);CHR$(23+64);CHR$(1);CHR$(23+64)
;CHR$(1);CHR$(25+64);"     ";CHR$(1)
;CHR$(22+64);" TRYK !";CHR$(1);CHR$
(22+64);
1250 LOCATE 2,8:PRINTCHR$(1);CHR$(2
2+64);" TRYK P for";CHR$(1);CHR$(22
+64);"     ";CHR$(1);CHR$(26+64);CHR
$(1);CHR$(25+64);"RETURN";CHR$(1);C
HR$(22+64);
1260 LOCATE2,9:PRINTCHR$(1);CHR$(22
+64);" 1/4 pause "CHR$(1);CHR$(22+6
4);"      ";CHR$(1);CHR$(22+64);" FO
R  ";CHR$(1);CHR$(22+64);
1270 LOCATE2,10:PRINTCHR$(1);CHR$(2
6+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(23+64);CHR$
(1);CHR$(23+64);CHR$(1);CHR$(23+64)
;CHR$(1);CHR$(23+64);CHR$(1);CHR$(2
3+64);CHR$(1);CHR$(23+64);CHR$(1);C
HR$(23+64);CHR$(1);CHR$(23+64);CHR$
(1);
```

```
1280 LOCATE12,10:PRINTCHR$(1);CHR$(
23+64);CHR$(1);CHR$(23+64);CHR$(1);
CHR$(27+64);"       ";CHR$(1);CHR$(22
+64);" SLUT ";CHR$(1);CHR$(22+64);
1290 LOCATE20,11:PRINTCHR$(1);CHR$(
26+64);CHR$(1);CHR$(23+64);CHR$(1);
CHR$(23+64);CHR$(1);CHR$(23+64);CHR
$(1);CHR$(23+64);CHR$(1);CHR$(23+64
);CHR$(1);CHR$(23+64);CHR$(1);CHR$(
27+64);
1300 LOCATE 4,1:PRINT" BASIC PLAY P
ROGRAM"
1310 LOCATE 2,2:PRINTCHR$(219);CHR$
(219);CHR$(219);CHR$(219);CHR$(219)
;CHR$(219);CHR$(219);CHR$(219);CHR$
(219);CHR$(219);
1320 LOCATE12,2:PRINTCHR$(219);CHR$
(219);CHR$(219);CHR$(219);CHR$(219)
;CHR$(219);CHR$(219);CHR$(219);CHR$
(219);CHR$(219);CHR$(219);CHR$(219)
;CHR$(219);CHR$(219);CHR$(219);
1330 RETURN
1340 PLAY "M4000S0L8CDEFGAB":PLAY "
R8"
1350 PLAY "V13","V13","V13"
1360 PLAY "L8CDEFGAB":PLAY "R8"
1370 PLAY "O4L4C","O4L4E","O4L4G":P
LAY"R4"
1380 PLAY "O3B","O4D","O4F":PLAY "R
4"
1390 PLAY "O4C","O4E","O4G"
1400 RETURN
```





Demonstration af Basic Play

```
30010  PLAY"L8V13"
30020  PLAY"O4E"
30030  PLAY"O4E"
30040  PLAY"O4E"
30050  PLAY"O4E"
30060  PLAY"O4G"
30070  PLAY"O4E"
30080  PLAY"O4E"
30090  PLAY"O4E"
30100  PLAY"O4G"
30110  PLAY"O4E"
30120  PLAY"O4C"
30130  PLAY"O4E"
30140  PLAY"O4D"
30150  PLAY"R"
30160  PLAY"R"
30170  PLAY"O4D"
30180  PLAY"O4D"
30190  PLAY"O4D"
30200  PLAY"O4D"
30210  PLAY"O4F"
30220  PLAY"O4D"
30230  PLAY"O4D"
30240  PLAY"O4D"
30250  PLAY"O4C"
30260  PLAY"O4D"
30270  PLAY"O4E"
30280  PLAY"O4D"
30290  PLAY"O4C"
30300  PLAY"R"
30310  PLAY"R"
30320  PLAY"O5C"
30330  PLAY"R"
30340  PLAY"O4G"
30350  PLAY"O4G"
30360  PLAY"O4G#"
30370  PLAY"O4G"
30380  PLAY"R"
30390  PLAY"O4B"
30400  PLAY"O5C"
```

Dette var lidt af stykket
Se den lille kattekilling

BASIC MED PETER.

Den forrige opgave var måske lidt svær, hvis ikke man før har stiftet bekendtskab med READ-DATA rutinen.

Især kunne det være svært at få programmet til at køre, hvis man ikke kunne gennemskue, at tallet nul blev brugt som stopklods, d.v.s., at når 0 i datasætningen blev nået, så var programmet slut.

Med linie 30 blev der undersøgt om værdien af A og B var nul. Den ene fejl lå her:

```
30 IF A = 0 OR B = 0 .....
```

- ville være den rigtige undersøgelse af A og B.

Den næste fejl lå i linie 70, hvor der sker en sammenligning af summem af A og B (= D) og det indtastede tal var ens. Det indtastede tal lå i variablen C og ikke A.

```
70 IF D = C THEN ....
```

- er korrekt. Her er iøvrigt et godt eksempel på udnyttelse af IF-THEN-ELSE strukturen, som vi tidligere har berørt.

Den sidste fejl ligger i linie 90. Vi arbejder med talvariabler og der ikke må forekomme bogstaver iblandt. Derfor skal A erstattes med et tal.

Lad os nu se på Strengvariabler. Det er dem der bruges til bogstaver og meget andet godt En strengvariabel kan se således ud: A$ - og have et indhold af tal, bogstaver eller begge dele. Eks.: A$ = "Dette er et tal: 8"

Alle de funktioner vi har gennemgået indtil nu kan uden videre anvendes til strengvariable.

Programmet til næste nummer stiller måske lidt store krav, men fortvivl ikke. Det er vigtigt at have styr på disse ting for at kunne lave sine egne programmer.

Opgaven lyder:

Lav et program, der spørger om hovedstaden i forskellige lande. Giv point for rigtige svar, og meddel om svaret var korrekt.

Løsningen kan gribes an på forskellig måde, men lad os bruge de metoder vi hidtil har arbejdet med. Løsningsforslaget er programmeret mere struktureret end vi tidligere har set, og hvis du vænner dig til det, når du laver dine egne programmer, vil de blive mere overskuelige - især når det gælder større programmer.

Der er indlagt 3 fejl i nedstående løsningsforslag:

```
10 WIDTH 40
20 READ A$
30 IF A$ = B$ THEN GOTO 100
40 PRINT "HVAD HEDDER HOVEDSTAD-
   EN I "; A$ ;"?"
50 INPUT B$
60 READ B$
70 IF B$ < > A$ THEN GOSUB 120
80 IF B$ = A$ THEN GOSUB 130
90 GOTO 20
100 PRINT "SPILLET ER FORBI, DU
    FIK "; Y ; "POINT"
110 END
120 PRINT "SVARET ER FORKERT":RE
    TURN
130 PRINT "SVARET ER KORREKT"
140 X=X+1
150 PRINT "DU HAR "; X "POINT"
160 RETURN
170 DATA DANMARK, KØBENHAVN, VES
    TTYSKLAND, BONN, ENGLAND, LO
    NDON, ITALIEN, ROM, *
```

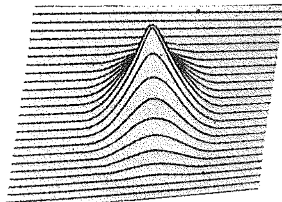
God jagt med fejlsøgningen.

peter

REGRESSION AF VITUS NILSEN

Her et et lille (!) program jeg længe har haft liggende i BIB. Men som jeg synes fortjener at blive bragt i bladet. Programmet hedder 'REGRESSION', og er skrevet af J. Vitus Nielsen fra byernes by, Rødovre. Programmet bruges til at tilnærme grafer ud fra indsamlede data. Det er et alle tiders redskab for alle matematikere, og er rigelig indtastningsbesværet værd.

Programmet er skrevet til MSX, men efter programmet har jeg lavet rettelser til SVI 318/328.



```
10 ' REGRESSION MSX-version 1.0
20 ' J. Vitus Nielsen 1988
30 '
40 CLS:KEY OFF:SCREEN 0:DEFSNGA-Z
50 LOCATE 5,5:PRINT"1. KORRELATIONSKOEFFIC
IENT"
60 LOCATE 5,8:PRINT"2. LINEÆR REGRESSION"
70 LOCATE 5,11:PRINT"3. MULTIPEL LINER REG
RESSION"
80 LOCATE 5,14:PRINT"4. N'te GRADS REGRESS
ION"
90 LOCATE 5,17:PRINT"5. EKSPONENTIEL REGRE
SSION"
100 LOCATE 5,20:PRINT"6. SLUT"
110 LOCATE 5,25:PRINT"VÆLG UNDERPROGRAM (1
-6)";
120 INPUTK1
130 IFK1<1ORK1>6GOTO 110
140 ONK1GOTO160,320,600,1280,2000
150 END
160 CLS
170 PRINT"KORRELATIONSKOEFFICIENT"
180 PRINT"=======================":PRINT:P
RINT
190 INPUT "ANTAL KOORDINATER";N
200 J=0:K=0:L=0:M=0:R=0
210 FORI=1TON
220 PRINT"PUNKT";I;"KOORDINATER (X,Y) ";
230 INPUTX,Y
240 J=J+X:K=K+Y:L=L+X^2:M=M+Y^2:R=R+X*Y
250 NEXT
260 R2=(N*R-J*K)/(SQR(N*L-J^2)*(N*M-K^2))
270 PRINT:PRINT"KOEFFICIENTEN ER LIG ";R2
280 PRINT:PRINT"VIL DU KØRE PROGRAMMET IGE
N? (1=JA, 0=NEJ)";
290 INPUTD:PRINT
300 IFD=1THENGOTO 190
310 GOTO 40
320 CLS:PRINT"LINEÆR REGRESSION":RG=1
330 PRINT"=================":PRINT
340 RG=1:INPUT"ANTAL KOORDINATER ";N
350 DIMPX(N),PY(N)
360 J=0:K=0:L=0:M=0:R2=0
370 FORI=1TON
380 PRINT"PUNKT";I;"KOORDINATER (X,Y) ";:I
NPUTX,Y:PX(I)=X:PY(I)=Y
390 J=J+X:K=K+Y:L=L+X^2:M=M+Y^2
400 R2=R2+X*Y:NEXTI
410 B=(N*R2-K*J)/(N*L-J^2):A=(K-B*J)/N
420 PRINT:PRINT"F(X) =";A;"+ (";B;"* X)"
430 J=B*(R2-J*K/N):M=M-K^2/N:K=M-J
440 R2=J/M:PRINT
450 PRINT"FORKLARET VARIATION (R^2) = ";
460 PRINTUSING"#.###";R2
470 PRINT"  KORRELATIONSKOEFFICIENT = ";
480 PRINTUSING"#.###";SQR(R2)
490 PRINT"      ESTIMERET SPREDNING = ";
500 PRINTUSING"#.###";SQR(K/(N-2))
510 PRINT:PRINT"INTERPOLATION:"
520 PRINT"(INDTAST 0  HVIS SLUT ØNSKES)"
530 PRINT"(INDTAST .1 HVIS GRAF ØNSKES)"
540 PRINT"INDTAST X ";:INPUTX
550 IFX=0THEN40
560 IFX=.1THEN2750
570 PRINT"          Y =";A+B*X
580 PRINT:GOTO 540
590 END
600 CLS:PRINT"MULTIPEL LINEÆR REGRESSIION"
610 PRINT"==========================="
620 INPUT "         ANTAL KOORDINATER ";N
630 INPUT "ANTAL UAFHÆNGIGE VARIABLER";V
640 PRINT:DIM X(V+2),S(V+2),T(V+1),A(V+1,V
+2)
650 X(1)=1:FORI=1TON:PRINT
660 PRINT "PUNKT";I
670 FOR J=1TOV
680 PRINT"UAFHÆNGIG VARIABEL";J;
690 INPUTX(J+1):NEXTJ
700 PRINT"  AFHÆNGIG VARIABEL ";
710 INPUT X(V+2)
720 FORK=1TOV+1
```

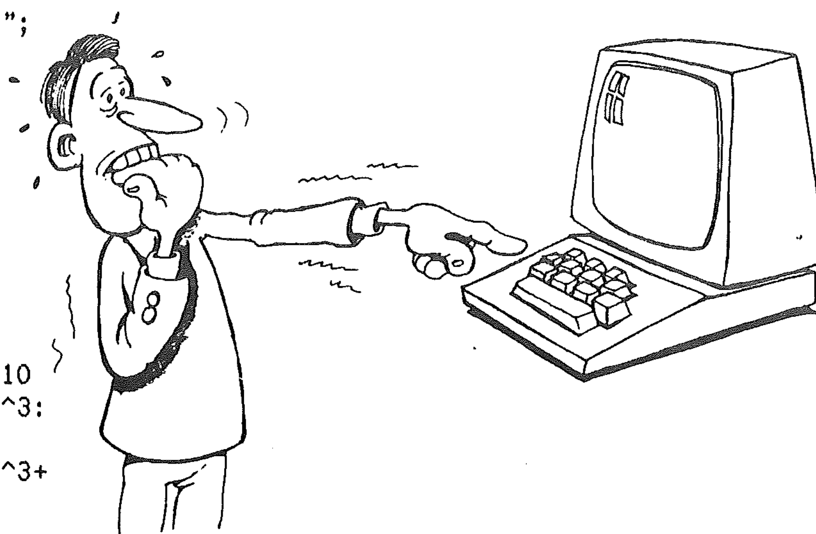
```
730 FORL=1TOV+2
740 A(K,L)=A(K,L)+X(K)*X(L)
750 S(K)=A(K,V+2)
760 NEXTL:NEXTK
770 S(V+2)=S(V+2)+X(V+2)^2
780 NEXTI
790 FORI=2TOV+1
800 T(I)=A(1,I)
810 NEXTI
820 FORI=1TOV+1:J=I
830 IFA(J,I)<>0THENGOTO 880
840 J=J+1
850 IFJ<=V+1THENGOTO 830
860 PRINT:PRINT "!!! INGEN ENTYDIG LØSNING
"
870 PRINT:GOTO1270
880 FORK=1TOV+2
890 B=A(I,K)
900 A(I,K)=A(J,K)
910 A(J,K)=B:NEXTK
920 Z=1/A(I,I)
930 FOR K=1TOV+2
940 A(I,K)=Z*A(I,K):NEXTK
950 FORJ=1TOV+1
960 IFJ=ITHENGOTO1010
970 Z=-A(J,I)
980 FORK=1TOV+2
990 A(J,K)=A(J,K)+Z*A(I,K)
1000 NEXTK
1010 NEXTJ
1020 NEXTI
1030 PRINT:PRINT"LIGNINGENS KOEFFICIENTER:
":PRINT
1040 PRINT"  KONSTANT =";A(1,V+2)
1050 FORI=2TOV+1
1060 PRINT"VARIABEL";I-1;"=";A(I,V+2)
1070 NEXTI:P=0
1080 FORI=2TOV+1
1090 P=P+A(I,V+2)*(S(I)-T(I)*S(1)/N)
1100 NEXTI
1110 R=S(V+2)-S(1)^2/N
1120 Z=R-P:L=N-V-1:PRINT:I=P/R
1130 'STOP
1140 PRINT"FORKLARET VARIATION (R^2) =";I
1150 PRINT"  KORRELATIONSKOEFFICIENT =";SQ
R(I)
1160 PRINT"        ESTIMERET SPREDNING =";SQ
R(ABS(Z/L)):PRINT
1170 PRINT"INTERPOLATION:"
1180 PRINT"(INDTAST 0 HVIS SLUT ØNSKES)"
1190 P=A(1,V+2)
1200 FORJ=1TOV
1210 PRINT"        VARIABEL";J;:INPUTX
1220 IFX=0THEN GOTO1270
1230 P=P+A(J+1,V+2)*X
1240 NEXTJ
1250 PRINT"AFHÆNGIG VARIABEL =";P:PRINT
1260 GOTO1190
1270 END
1280 CLS:PRINT"N'te GRADS REGRESSION":RG=2
1290 PRINT"====================":PRINT
1300 RG=2:INPUT "   LIGNINGENS GRAD ";D
1310 INPUT " ANTAL KOORDINATER ";N
1320 DIM A(2*D+1),R(D+1,D+2),T(D+2),PX(N),
PY(N),B(D+1)
1330 A(1)=N
1340 FORI=1TON
1350 PRINT"PUNKT";I;"KOORDINATER (X,Y) ";
1360 INPUT X,Y:PX(I)=X:PY(I)=Y
1370 FOR J=2TO2*D+1
1380 A(J)=A(J)+X^(J-1):NEXTJ
1390 FORK=1TOD+1
1400 R(K,D+2)=T(K)+Y*X^(K-1)
1410 T(K)=T(K)+Y*X^(K-1):NEXTK
1420 T(D+2)=T(D+2)+Y^2
1430 NEXTI
1440 FORJ=1TOD+1
1450 FORK=1TOD+1
1460 R(J,K)=A(J+K-1):NEXTK,J
1470 FOR J=1TOD+1
1480 K=J
1490 IFR(K,J)<>0THEN1540
1500 K=K+1
1510 IFK<+D+1THEN1490
1520 PRINT:PRINT"!!! INGEN ENTYDIG LØSNING
":PRINT
1530 GOTO40
1540 FORI=1TOD+2
1550 S=R(J,I)
1560 R(J,I)=R(K,I):R(K,I)=S
1570 NEXTI
1580 Z=1/R(J,J)
1590 FORI=1TOD+2
1600 R(J,I)=Z*R(J,I):NEXTI
1610 FORK=1TOD+1
1620 IFK=JTHEN1670
1630 Z=-R(K,J)
1640 FORI=1TOD+2
1650 R(K,I)=R(K,I)+Z*R(J,I)
1660 NEXTI
1670 NEXTK
1680 NEXTJ:PRINT
1690 PRINT"                    KONSTANT = ";
1700 PRINTUSING"###.###";R(1,D+2):A=R(1,D+
2)
1710 FORJ=1TOD
1720 PRINT" ";J;"'TE GRADS KOEFFICIENT = "
;:PRINTUSING"###.###";R(J+1,D+2):B(J)=R(J+
1,D+2)
1730 NEXTJ:PRINT
```

```
1740 P=0
1750 FORJ=2TOD+1
1760 P=P+R(J,D+2)*(T(J)-A(J)*T(1)/N)
1770 NEXTJ
1780 Q=T(D+2)-T(1)^2/N
1790 Z=Q-P:I=N-D-1:PRINT
1800 J=P/Q
1810 PRINT"FORKLARET VARIATION (R^2) = ";
1820 PRINTUSING"###.###";J
1830 PRINT" KORRELATIONSKOEFFICIENT = ";
1840 PRINTUSING"###.###";SQR(J)
1850 PRINT"      ESTIMERET SPREDNING = ";
1860 PRINTUSING"###.###";SQR(ABS(Z/I))
1870 PRINT:PRINT"INTERPOLLATION :"
1880 PRINT:PRINT"INDTAST '0' HVIS SLUT ØNS
KES"
1890 PRINT"INDTAST 0.1 HVIS GRAF ØNSKES":P
RINT
1900 P=R(1,D+2)
1910 INPUT"INDTAST X";X
1920 IFX=0THEN40
1930 IFX=.1THEN2750
1940 FORJ=1TOD
1950 P=P+R(J+1,D+2)*X^J
1960 NEXTJ
1970 PRINT"          Y=";P
1980 PRINT:GOTO 1900
1990 END
2000 CLS:PRINT"EKSPONENTIEL REGRESSION":RG
=3
2010 PRINT"======================="
2020 RG=3:PRINT:INPUT"          ANTAL KOORDI
NATER ";N
2030 DIMPX(N),PY(N):J=0:K=0:L=0:M=0:R2=0
2040 FORI=1TON
2050 PRINT"PUNKT";I;"KOORDINATER (X,Y) ";
2060 INPUTX,Y:PX(I)=X:PY(I)=Y
2070 Y=LOG(Y):J=J+X:K=K+Y:L=L+X^2:M=M+Y^2:
R2=R2+X*Y
2080 NEXTI
2090 B=(N*R2-K*J)/(N*L-J^2)
2100 A=(K-B*J)/N:PRINT
2110 PRINT "A =";:PRINTUSING"###.###";EXP(
A)
2120 PRINT "B =";:PRINTUSING"###.###";B
2130 J=B*(R2-J*K/N):M=M-K^2/N:K=M-J:PRINT
2140 R2=J/M
2150 PRINT"FORKLARET VARIATION (R^2) = ";
2160 PRINTUSING"###.###";R2
2170 PRINT" KORRELATIONSKOEFFICIENT = ";
2180 PRINTUSING"###.###";SQR(R2)
2190 PRINT"      ESTIMERET SPREDNING = ";
2200 PRINTUSING"###.###";SQR(K/(N-2))
2210 PRINT:PRINT"INTERPOLATION:"
2220 PRINT:PRINT"INDTAST'0' HVIS SLUT ØNSK
ES"
2230 PRINT"INDTAST 0.1 HVIS GRAF ØNSKES"
2240 PRINT:PRINT "INDTAST X";
2250 INPUTX
2260 IFX=0THEN 40
2270 IFX=.1THEN2750
2280 PRINT "          Y=";EXP(A)*EXP(B*X)
2290 PRINT:GOTO 2240
2300 END
2310  COLOR 15,4,4:WIDTH39
2320 CLS:PRINT
2330 PRINT" Trykker du 'ENTER' når du ska
l skriveen værdi, bruger programmet de på
for-hånd indsatte værdier (dem i klammerne
)"
2340 PRINT" Når grafen er tegnet kan du æ
ndre  påkoordinatsystemet  ved at taste 'N
'  ogreturnere til delprogrammet ved at ta
s-te 'E'."
2350 A$=INPUT$(1)
2360 CLS:KEYOFF:COLOR 15,4
2370 TJ=0:SCREEN0
2380 COLOR15,1:CLS:KS$=CHR$(91):KE$=CHR$(9
3)
2390 PRINT"x start= "KS$"-10"KE$TAB(26);:L
INEINPUTA$:IFA$=""THENXS=-10ELSEXS=VAL(A$)
:IFXS>-1THEN2390
2400 PRINT"x slut= "KS$"10"KE$TAB(26);:LIN
EINPUTA$:IFA$=""THENXE=10ELSEXE=VAL(A$):IF
XE<1THEN2400
2410 PRINT"f start= "KS$"x start"KE$TAB(26
);:LINEINPUTA$:IFA$=""THENFS=0ELSEFS=VAL(A
$):LINEINPUT"å(b)en / (l)ukket ";A$:IFA$="
B"ORA$="b"THENSU=1ELSESU=2
2420 PRINT"f slut= "KS$"x slut"KE$TAB(26);
:LINEINPUTA$:IFA$=""THENFE=0ELSEFE=VAL(A$)
:LINEINPUT"å(b)en / (l)ukket ";A$:IFA$="B"
ORA$="b"THENEU=1ELSEEU=2
2430 AF=INT(191/(XE-XS))
2440 PRINT:PRINT"y kordinat for centrum="T
AB(26);:LINEINPUTA$:IFA$=""THENYK=96/AFELS
EYK=VAL(A$)
2450 PRINT:PRINT"graf helt ud (j/n) "KS$"j
"KE$TAB(26);:LINEINPUTA$:IFA$="n"ORA$="N"T
HENHU=0ELSEHU=1
2460 PRINT:PRINT"net (j/n) "KS$"j"KE$TAB(2
6);:LINEINPUTA$:IFA$="n"ORA$="N"THENNE=1EL
SENE=0
2470  COLOR15,4,4:SCREEN 2
2480 OPEN"grp:"AS#1
2490 AF=INT(191/(XE-XS))
2500 KX=INT(ABS(XS)/(XE-XS)*191):KY=INT(YK
*AF):LINE(KX,0)-(KX,191),15:LINE(0,KY)-(19
1,KY),15
```

```
2510 X=KX-INT(KX/AF)*AF:Y=KY-INT(KY/AF)*AF
2520 IFHU=1THENS1=-KX/AF:E1=(191-KX)/AFELS
EE1=XE:S1=XS
2530 IFNE=0THENFORR=YTO191STEPAF:FORT=XTO1
91STEPAF:PSET(T,R),15:NEXTT,R
2540 FORT=XTO191STEPAF:LINE(T,KY-1)-(T,KY+
1),15:NEXT
2550 FORT=YTO191STEPAF:LINE(KX-1,T)-(KX+1,
T),15:NEXT
2560 IFFS=0THENFS=S1
2570 IFFE=0THENFE=E1
2580 IFSU>1THENPSET(INT(KX+FS*AF-.5),INT(K
Y-FNF(FS)*AF-.5)),15:DRAW"r2d2l2u2":IFSU=1
THENPRESETSTEP(1,1)ELSEPSETSTEP(1,1),15
2590 FORX=FSTOFESTEP1/AF
2600 F=FNF(X)*AF
2610 IFTJ=0ORT1=1THENPSET(INT(KX+X*AF+.5),
INT(KY-F+.5)),15:TJ=1
2620 IFT1=0THENLINE-(INT(KX+X*AF+.5),INT(K
Y-F+.5)),15
2630 NEXT
2640 IFEU>1THENPSETSTEP(-1,-1),15:DRAW"r2d
2l2u2":IFEU=1THENPRESETSTEP(1,1)ELSEPSETST
EP(1,1),15
2650 LINE(0,0)-(191,0),4:LINE(0,191)-(191,
191),4
2660 FOR I=1TON
2670 PSET(INT(KX+PX(I)*AF)-1,INT(KY-PY(I)*
AF)-1),4:PRINT#1,CHR$(211)
2680 NEXT
2690 A$=INPUT$(1)
2700 IFA$="n"ORA$="N"THENTJ=0:SCREEN0:CLOS
E#1:GOTO2390
2710 IFA$="e"ORA$="E"THENCLOSE#1:GOTO2830
2720 GOTO2690
2730 X=X+1/AF:TJ=0:IFX>FETHEN2640ELSERESUM
E2600
2740 RETURN2370
2750 CLS:PRINT"DU ØNSKER EN GRAF? (J/N)";
2760 INPUTA$
2770 IFA$="J"OR A$="j"THEN2800
2780 IFA$="N"OR A$="n"THEN40
2790 GOTO2750
2800 ONRGGOTO2810,2840,2890
2810 DEFFNF(X)=A+B*X
2820 GOTO2310
2830 ONRGGOTO350,1300,2020
2840 ONDGOTO2850,2860,2870,2880
2850 DEFFNF(X)=A+B(1)*X:GOTO2310
2860 DEFFNF(X)=A+B(1)*X+B(2)*X^2:GOTO2310
2870 DEFFNF(X)=A+B(1)*X+B(2)*X^2+B(3)*X^3:
GOTO2310
2880 DEFFNF(X)=A+B(1)*X+B(2)*X^2+B(3)*X^3+
B(4)*X^4:GOTO2310
2890 DEFFNF(X)=EXP(A)*EXP(B*X)
2900 GOTO2310
65000 '&&1 Tjeksum
65010 'T
65020 '111YX SWPRG QZ77P KA2LB OKUCR
65030 'I6NBX UBYK1 3ORTC 35DMX 6PA3Y
65040 'YT266 Z28PK JHXGB V4IVN 6AKW6
65050 '7VMA3 ME52G BGI8I U9I98 ARIHQ
65060 'OM3VA XLAQM UU111 6CVR4 N6UOW
65070 'SPGEL Q7POK UYJ7Y 9FM57 W58L6
65080 'L3UBW BMTBW 7U4BJ MQAO4 1GAS9
65090 'ROXSL XDPB3 QT8OX P6Z13 OFVPD
65100 '482OK UXMJX UWGRX EP63Y Y541J
65110 '6Z2QP 7OZG3 XE8GF 6ENGQ 6LQNV
65120 '6HGBE NKNV2 JMR19 OSR3Z MEG9E
65130 'BIS8J NAL74 D1SUA
```

Rettelser til SVI 318/328:

```
40 CLS:SCREEN 0,0:DEFSNGA-Z
2360 CLS:COLOR 15,4
2370 TJ=0:SCREEN0,0
2470  COLOR15,4,4:SCREEN 1
2480 '
2670 PSET(INT(KX+PX(I)*AF)-1,INT(KY-PY(I)*
AF)-1),4:PRINTCHR$(211)
2700 IFA$="n"ORA$="N"THENTJ=0:SCREEN0:GOTO
2390
2710 IFA$="e"ORA$="E"THEN2830
```



ORMEN DEN LANGE.

Dette program viser en orm der kravler hen over skærmen. En nærmere beskrivelse findes i Scientific American nummer 12 1987 side 98-102. Ormen dannes af en række cirkler, og den synes at bevæge sig fordi der tilføjes en cirkel i den ene ende og slettes en i den anden. Tabellerne XCIRC og YCIRC husker cirklernes placering på skærmen, så at den bageste kan blive slettet. TAIL angiver hvilken plads i tabellen der svarer til ormens ende. Ved at bruge modulusfunktionen kommer plads nummer 1 til at følge efter plads nummer 25 (lin.460). Den sidste cirkel slettes i lin.470, og den nye tegnes i lin.600. Ormen må ikke lave for bratte bevægelser, det ser unaturligt ud. Så man kan ikke bare lægge RND til X- og Y-værdierne. DIR angiver kravleretningen, og den kan kun variere plus/minus 10 grader, svarende til 0.1745 radianer (lin. 490). Dette omsættes så til forandringen i X- og Y-værdierne i lin. 520-530. Hvis ormen kravler ud over den ene side af skærmen, kommer den til syne fra den anden side (lin. 540-570).

Ofte kan man gøre et program hurtigere ved at tilpasse det til maskinen det skal køre på. Det tager meget lang tid at beregne sinus og cosinus, og her er der ikke brug for den store nøjagtighed med mange decimaler. Så i stedet nøjes jeg med 16 værdier, der kan slås op i tabellerne SI og CO (lin.40-90). AND 15 sørger for at index i tabellerne holder sig inden for 0-15. Den samme bit-aritmetik klarer at ændre tallet -1 til 255, så ormen skifter side på skærmen. Som integer er -1 nemlig lutter binære ettaller, hvoraf vi tager de otte længst til højre med AND 255 (lin.190). Da 191 ikke kun består af ettere, må der bruges to linier på at holde ormen på skærmen i det lodrette plan (lin.200-210). Det betyder nok mindre for hastigheden at jeg har indført variablerne GX og GY, der står for gammel X og gammel Y. Så skal der kun slås op i XC og YC en enkelt gang i lin.150 i stedet for både i lin.160 og 170. Prøv selv at sætte fut i andre programmer !

```
5 ' Ormeprogram med større hastighed
10 DEFINT A-Z: DEFSNG D
20 DIM XC(25), YC(25)
30 DIM SI(15), CO(15)
40 FOR N = 0 TO 15: READ SI(N): NEXT
50 DATA 0, 2, 3, 4, 4, 4, 3, 2
60 DATA 0,-2,-3,-4,-4,-4,-3,-2
70 FOR N = 0 TO 15: READ CO(N): NEXT
80 DATA  4, 4, 3, 2, 0,-2,-3,-4
90 DATA -4,-4,-3,-2, 0, 2, 3, 4
100 X = 100: Y = 100 'Startpunkt
110 DI = 0: TA = 0
120 DUMMY = RND(-TIME)
130 COLOR 1,15: SCREEN 2
140 TA = (TA+1) MOD 26
150 GX = XC(TA): GY = YC(TA)
160 CIRCLE(GX,GY),3,15
170 PSET(GX,GY),9   'Slim !
180 DI = DI + (RND(1)-.5)*2
190 X = (X+CO(DI AND 15)) AND 255
200 Y = (Y+SI(DI AND 15)) MOD 192
210 IF Y < 0 THEN Y = Y + 191
220 CIRCLE(X,Y),3,1
230 XC(TA) = X: YC(TA) = Y
240 IF INKEY$ = "" THEN 140
250 SCREEN 0
300 PRINT "Originaludgaven kommer nu ..."
310 PRINT: PRINT "Tryk på en tast ";
320 S$ = INPUT$(1)
330 CLEAR
340 'Udgave næsten direkte skrevet
350 'af efter Scientific American
360 'nr 12/1987 side 101
370 DIM XCIRC(25), YCIRC(25)
380 XCIRC(1) = 100
390 YCIRC(1) = 100
400 DIR = 0
410 TAIL = 1
420 DUMMY = RND(-TIME)
430 COLOR 1,15: SCREEN 2
440 'Repeat
450 WASTAIL = TAIL
460 TAIL = (TAIL MOD 25) + 1
470 CIRCLE(XCIRC(TAIL),YCIRC(TAIL)),3,15
480 CHANGE = RND(1)
490 IF CHANGE < .5 THEN DIR = DIR+.1745
    ELSE DIR = DIR-.1745
500 X = XCIRC(WASTAIL)
510 Y = YCIRC(WASTAIL)
520 NX = X + 4 * COS(DIR)
530 NY = Y + 4 * SIN(DIR)
540 NX = NX MOD 256 '600 på PC
550 NY = NY MOD 192 '200 på PC
560 IF NX < 0 THEN NX = NX + 256
570 IF NY < 0 THEN NY = NY + 192
580 XCIRC(TAIL) = NX
590 YCIRC(TAIL) = NY
600 CIRCLE(NX,NY),3,1
610 'Until key pressed
620 IF INKEY$ = "" THEN GOTO 450
630 SCREEN 0
640 END
```

SVENSKA SIDAN

Så var det dags för ytterligare en del av vår maskinspråksskola. Den här gången kommer vi att gå igenom pekare. Pekare är inte så svårt som vissa tror. En pekare brukar oftast vara ett registerpar. Detta par brukar vara HL. HL gives ett adressvärde tex D050 (HL = D050 eller H = D0 och L = 50). Eftersom HL är variabelt så kan HL antaga vilka värden som helst.
En liten parantes för er som undrar vad en adress är så kommer här en liten förklaring på detta. Minnet består av ett antal adresser. Varje adress innehåller ett åttabitars tal. Talet kan antingen vara ett kommado eller innehålla ett datavärde. Slut parantes.
Vi sa tidigare att HL pekade på adressen D050. På denna adress ligger då ett data värde. Detta datavärde kan sedan läggas i register A (ackumulatorn) med hjälp av ett kommando som vi kommer att ta upp lite senare. Värdet kan sedan skrivas ut på skärmen med hjälp av OUTDO.

Vi kommer också så smått att gå in på något som heter flaggor. De mest användbara flaggorna är:

Z = Nollflagga,Zeroflagga
C = Carry

Dessa flaggor används som villkor vid hopp. Eftersom en flagga bara ligger på en bit kan dess värde bara vara 1 eller 0. 1 när det är SANT och 0 när det är FALSKT. Närmare förklaring kommer efter hand i denna kurs.

De nya kommandon vi nu kommer att gå igenom är:

LD HL,nn — Ladda HL med ett 16-bitars tal. Tex LD HL,D050.

LD A,(HL) — A laddas med det värde som HL pekar på dvs om D050 innehåller 5 får A detta värde.

CP n — jämför register A med det som sätts in i stället för n. Om A=n får flaggan Z värdet 1. Om A<n får flaggan C värdet 1 och om A>=n får flaggan C värdet 0.

RET Z — utför kommadot RET endast då Z=1.

INC HL — ökar (increase) HL:s värde med ett. Om HL = D050 blir HL = D051 efter INC HL.

JP nn — Hoppar till adressen nn.

Mycket nytt men här går det fort framåt. Om du inte förstår det nu så lär du nog förstå det efter vårt programexempel.

Nu till det utlovade programmen:

Först BASIC-programmet som matar in den önskade raden:

```
10 CLS
20 LINEINPUT A¤
30 FOR I=1 TO LEN(A¤)
40 POKE &HD04F+I,ASC(MID¤(A¤,I,1))
50 NEXT I
60 POKE &HD04F+I,255
```

När ni startar detta program kommer en tom skärm upp med en markör i det övre vänstra hörnet. På denna rad skriver ni in ett meddelande och trycker sedan på ENTER. Programmet översätter nu bokstäverna, som ni skrivit in, till ASCII-koder som i sin tur läggs i adresserna D050 och framåt (beroende på längden på det som du skrivit in). Allra sist lägger vi in ett kontrolltecken som talar om för maskinspråksprogrammet när textraden är slut. Spara nu detta program efter det att ni kört det. Skriv sedan NEW.

Och nu assemblerprogrammet:

```
D000: LD HL,D050
D003: LD A,(HL)
D004: CP FF
D006: RET Z
D007: CALL OUTDO
D00A: INC HL
D00B: JP D003
```

Det ni ser längst till vänster är adresserna som de olika kommandona hamnar på. Anledningen till att vi visar dem är att vi har med en hoppinstruktion (JP D003), som hoppar till adressen D003. För er som har assemblator är vi nog närmare en korrekt beskrivning av programmet om vi skriver så här:

```
         LD HL,D050
LABEL :  LD A,(HL)
         CP FF
         RET Z
         CALL OUTDO
         INC HL
         JP LABEL
```

Vad gör då det här programmet? Jo om vi börjar från början, vilket brukar vara den bästa lösningen på alla problem, så kan vi där konstatera att vi ger registerparet HL värdet D050. D050 är i det här fallet den adress där datan för er inskrivna text startar. Det som sedan sker i kommandot LD A,(HL) är att registret A får det värde som HL pekar på. Detta värde är alltså inte D050, utan en ASCIIkod från er inskrivna text, tex C3. Sen hittar vi CP FF. Vad är nu detta då? Jo i slutet av ert textmeddelande satte vi ju ett kontrolltecken som faktiskt var FF (255). Det som sker här är att register A fick ett värde som kom från det som registerpar HL pekade på och om detta värde är FF så betyder det också att er textsnutt är slut och det är dags att återgå till BASIC:en. Som vi sa tidigare så blev Z flaggan lika med 1 om register A var lika med talet n (i detta fallet värdet FF), alltså har detta kommandot satt flaggan Z till ett om register A = FF. Sen kommer då RET Z! Javisst, om register A var 255 så var flaggan Z=1 och villkoret för att få komma tillbaka till BASIC:en är uppfyllt. Om villkoret ej är uppfyllt fortsätter programmet bara med att utföra nästa rad som är CALL OUTDO. Detta kommando känner vi igen sen förra kursen. Det skriver ut värdet (ASCIIkoden) i register A på skärmen. Vad har vi efter detta då? Jo INC HL, som är något nytt. Detta kommando ökade värdet på registerpar HL, som denna gång används som pekare, och talet som var D050 blir nu D051. Det betyder att nu pekar HL på bokstav nummer två i din skrivna text. HL kommer sedan givetvis att ökas så länge det finns bokstäver i ditt meddelande. Detta sköter ju CP FF till oss. Slut på förklaringar.

Om vi nu översätter vårt programexempel i assembler till maskinspråk mha vår lilla maskinkodslista, som ni förhoppningsvis har hunnit skaffa er vid det här laget, blir resultatet antagligen så här:

```
LD HL,D050  = 21,50,D0
LD A,(HL)   = 7E
CP FF       = FE,FF
RET Z       = C8
CALL OUTDO  = CD,18,00
INC HL      = 23
JP D003     = C3,03,D0
```

Det var alla nödvändiga koder och ni kan nu antagligen själv systemet för att hitta koderna till assemblerinstruktionerna. Detta betyder att vi i fortsättningen skriver rena assemblerprogram och ni blir därför tvingade att själv översätta dessa program (eller skaffa en assemblator).

För sista gången ges alltså här ett BASIC exempel som skriver in era maskinspråkskoder. Nu BASICprogrammet:

```
10 CLS : ADR=&HD000
20 READ A¤
30 IF A¤="XX" THEN END
40 POKE ADR,VAL("&H"+A¤)
50 ADR=ADR+1
60 GOTO 20
70 DATA 21,50,D0,7E,FE,FF
80 DATA C8,CD,18,00,23
90 DATA C3,03,D0,XX
```

När ni nu kört detta program och vill starta ert maskinspråk skriver ni:

```
DEFUSR=&HD000
Z=USR(0)
```

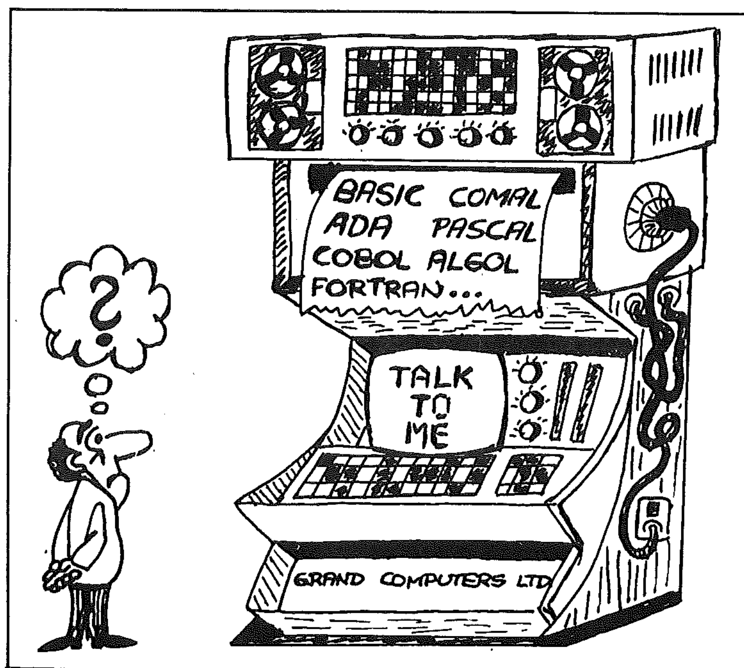
Vill ni spara era programsnuttar i framtiden (inräknat detta), skriver ni BSAVE "1:Namn",&hstartadr,&hslutadr,&hprogramstartsadr . Om ni ej har diskdrive så hoppa över 1: . I detta fallet blir det alltså:

```
BSAVE "1:Namn",&hD000,&HD00C,&HD000
```

Obs! Detta sparar alltså inte din inmatade textrad utan bara programmet som skriver ut den.

Det var allt för idag (allt och allt. Det var ganska långt faktiskt) och må era ovänners skrivare skriva med NLQ(Not Letter Quality).

MAGNUS OCH JOAKIM OCH TORNHILL OCH BALLDIN



Æ Ø Å PÅ SKÆRMEN.

```
10 'Dette program bruger hook HINIP
20 'til at definere de danske tegn
30 'hver gang SCREEN kaldes. Tegnene
40 'placeres i området SLTWRK, så de
50 'bevares indtil maskinen bliver
60 'slukket. SLTWRK bruges kun af ud-
70 'videlses-ROM.
100 DEFINT A-Z
110 'Hurtigere kørsel med integers
120 AD = &HFD30: NR = 80: GOSUB 180
130 AD = &HFDC7: NR =  3: GOSUB 180
140 SCREEN 0
150 PRINT "Dansk tegnsæt: ÆØÅ æøå"
160 END
170 'Subrutine poke
180 FOR AD=AD TO AD+NR
190   READ N: POKE AD,N
200 NEXT
210 RETURN
220 DATA 205,33,7,17,216,2,1,0,0,205
230 DATA 66,253,17,216,3,1,24,0,42,36
240 DATA 249,25,235,33,81,253,9,1,24
250 DATA 0,195,92,0,120,160,160,248
260 DATA 160,160,184,0,112,136,152
270 DATA 168,200,136,112,0,32,32,112
280 DATA 136,248,136,136,0,0,0,208,40
290 DATA 248,160,88,0,0,8,112,152,168
300 DATA 200,112,128,32,0,112,8,120
310 DATA 136,120,0
320 DATA 241,195,48,253
```

Hvis de danske tegn defineres inde fra Basic, holder de kun indtil man skifter skærmmode. Ovenstående program lader de danske bogstaver overleve, men de kan ikke vises på en grafikskærm, for ved skrivning på grafikskærmen tages tegnene altid fra ROM.
ÆØÅ og æøå ligger på tasterne [\] {|}, så de svarer til en almindelig printer. Det betyder at man skal bruge skiftenøglen for at skrive de små bogstaver !

LISTEBESKYTTELSE

I disse AIDS-tider er det blevet så moderne at beskytte både det ene og det andet. Her hopper jeg med på vognen med et program der kan listbeskytte et program.

Af: Kim Andersen

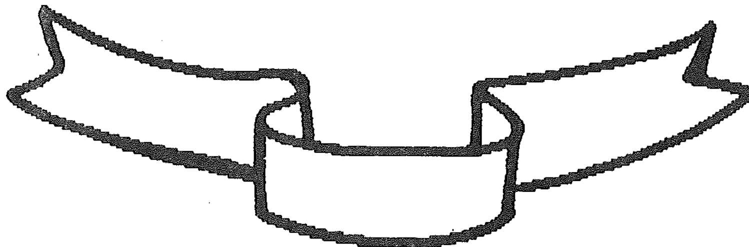
For et stykke tid siden fik jeg fra Tomas Abrahamsson i Sverige nogle programmer til BIB. Et af dem havde han listbeskyttet så man fik en fire-liniers tekst ud på skærmen når man forsøgte at liste programmet. Jeg lurede ideen af og har nu lavet en maskinkode rutine der kan listbeskytte et Basic-program efter samme koncept. Først taster man nedestående Basic-program ind. I linierne 100-130 står teksten man vil få skrevet ud på skærmen ved listning. Denne kan frit ændres efter behag (dog max. 30 tegn pr. linie). Efter indtastning gemmes evt. Basicprogrammet på bånd/disk under et eller andet navn, hvis man er et rigtigt sikkerhedsmenneske. Nu køres programmet, og er det korrekt indtastet (ellers bruger man TJEKSUM programmet), vil man på bånd/disk få gemt en fil "BESKYT.BIN" (på bånd "BESKYT"). Nu kan rutinen bruges. Har man et Basic-program man vil have beskyttet, skriver man 'BLOAD"BESKYT.BIN",R'. Prøver man nu at liste programmet, får man kun fire liniers tekst. Programmet kan fint RUN'es og man kan også fint SAVE'e det.

For at beskyttelsen skal virke ordentlig, skal man huske et par ting ved det Basic-program, man vil have beskyttet: Hvis programmet er lavet således at det på et tidspunkt går tilbage i Basic, skal man i sidste programlinie skrive 'xx POKE&H8100,0:POKE&H8101,0:END', ellers vil man igen kunne liste programmet. Har man et program der ikke går tilbage til Basic, skal man være opmærksom på at CTRL/STOP ikke virker, og man da må lave en kodeanordning, hvis man selv vil kunne gå ud af programmet.

Dette var ordene. Glædelig indtastning!

```
10 ' LIST-BESKYT MSX-version 1.0
20 ' Kim Andersen 1989
30 '
40 RESTORE100
50 AD=55585!
60 FORT=0TO3:READA$:A$=LEFT$(A$+SPACE$(30)
,30)
70 FORR=1TO30:POKEAD+R,ASC(MID$(A$,R,1)):N
EXTR
80 AD=AD+50
90 NEXTT
100 DATA Dette program er listbeskyttet
110 DATA af Kim Andersen efter ide
120 DATA af Tomas Abrahamsson
130 DATA Syntax'en 1989
140 '
150 RESTORE 220:J=0
160 FORI=&HD800TO&HD911
170 READX$:POKEI,VAL("&H"+X$)
180 J=J+VAL("&H"+X$):NEXTI
190 IFJ<>21965THENPRINT"DATAFEJL!":STOP
200 PRINT"Gør bånd/disk klar og tryk SPACE
":A$=INPUT$(1):IFA$<>" "THEN90 ELSE BSAVE"
BESKYT.BIN",&HD800,&HD9D9
210 '
220 DATA CD,41,00,F3,21,12,D9,0E
230 DATA 04,06,10,36,42,23,10,FB
240 DATA 06,1E,23,10,FD,36,C9,23
250 DATA 36,0D,23,36,0A,23,36,00
260 DATA 23,0D,20,E5,21,03,80,3E
270 DATA 05,BE,38,04,23,AF,BE,C8
280 DATA 21,AC,F3,11,94,D8,73,23
290 DATA 72,AF,CD,48,08,01,B3,D8
300 DATA 11,12,D9,60,69,3E,FF,BE
310 DATA 28,22,E5,CD,7B,52,E1,3E
320 DATA 13,85,6F,3E,00,8C,67,44
330 DATA 4D,62,6B,E5,CD,7B,52,E1
340 DATA 3E,32,85,6F,3E,00,8C,67
350 DATA 54,5D,18,D7,21,00,D9,CD
360 DATA 7B,52,21,06,D9,CD,7B,52
370 DATA 21,F8,F3,36,F8,23,36,FB
380 DATA 23,36,F0,23,36,FB,21,F0
390 DATA FB,36,0B,23,06,06,36,0D
400 DATA 23,10,FB,C9,21,00,80,06
410 DATA 00,3E,42,BE,20,02,36,7F
```

```
420 DATA 23,10,F6,AF,CD,48,08,21
430 DATA 00,81,36,00,23,36,00,CD
440 DATA 44,00,C9,31,20,50,4F,4B
450 DATA 45,26,48,46,42,42,30,2C
460 DATA 26,48,30,3A,27,00,32,20
470 DATA 50,4F,4B,45,26,48,46,42
480 DATA 42,31,2C,26,48,31,3A,27
490 DATA 00,33,20,50,4F,4B,45,26
500 DATA 48,38,31,30,30,2C,26,48
510 DATA 38,3A,27,00,34,20,50,4F
520 DATA 4B,45,26,48,38,31,30,31
530 DATA 2C,31,32,39,3A,27,00,FF
540 DATA 35,20,27,0D,0A,00,41,3D
550 DATA 55,53,52,39,28,30,29,0D
560 DATA 0A,00,00,00,00,00,00,00
65000 '&&1 Tjeksum
65010 'K
65020 '11134 TV29G 1KS19 RYEYV 1QJMO
65030 'V99VM XF66N K1P52 FAYSN LDYIB
65040 'TP5GH M
```



FÅ ORDEN PÅ VARIABLERNE

I MSX Basic kan variabelnavne være næsten lige så lange som man har lyst til, men det er kun de første to tegn der har betydning internt. Især når man indtaster programmer fra andre blade kan det give problemer. For eksempel forekom der i det originale ormeprogram variablerne NEWX og NEWY. De vil af MSX Basic blive betragtet som en enkelt variabel ved navn NE ! Måske kan det følgende lille maskinkodeprogram være til nytte. Det viser de interne navne på alle eksisterende variable. Der er temmelig mange tal at skrive ind, og jeg har ikke udregnet nogen checksum, så du må være omhyggelig! Til gengæld behøver det ikke at ligge på en bestemt adresse, sådan som mange maskinkodeprogrammer.

```
10 'Denne rutine viser de i øjeblikket
20 'eksisterende variable i den række-
30 'følge de blev skabt. Variablerne
40 'udskrives ordnet efter type.
50 'Skriv PRINT USR(0)
60 DEFINT A-Z
70 CLEAR 200,&HE000 'Rutinen kan pla-
80 'ceres næsten hvorsomhelst, bare
90 'der er plads til de 264 bytes.
100 AD = &HE000 : DEFUSR = AD
110 FOR AD = AD TO AD + 263
120 READ N : POKE AD,N
130 NEXT
200 DATA 33,225,233,34,246,247,205
210 DATA 246,247,229,221,225,17,193,0
220 DATA 25,221,117,63,221,116,64,221
230 DATA 117,82,221,116,83,17,58,0,25
240 DATA 34,196,254,17,248,255,25,34
250 DATA 194,254,62,13,223,62,10,223
260 DATA 253,42,196,254,253,54,3,4
270 DATA 175,71,253,119,0,42,194,246
280 DATA 237,91,196,246,231,48,9,205
290 DATA 202,224,79,9,24,246,24,232
300 DATA 42,196,246,237,91,198,246
310 DATA 231,48,68,205,202,224,35,32
320 DATA 54,35,175,71,253,42,196,254
330 DATA 253,119,1,126,253,119,2,135
340 DATA 79,9,253,126,1,167,62,40,40
350 DATA 2,62,44,253,119,1,223,86,43
360 DATA 94,229,235,205,18,52,225,43
370 DATA 253,42,196,254,253,53,2,32
380 DATA 223,62,41,223,43,86,43,94,25
390 DATA 35,35,24,181,253,42,196,254
400 DATA 253,126,0,167,40,10,62,13
410 DATA 223,62,10,223,253,42,196,254
420 DATA 42,194,254,35,34,194,254,253
430 DATA 53,3,32,144,62,3,50,99,246
440 DATA 33,11,0,34,248,247,201,126
450 DATA 221,42,194,254,221,190,0,35
460 DATA 245,32,34,221,78,4,253,42
470 DATA 196,254,253,126,0,167,253,54
480 DATA 0,1,40,6,62,44,223,62,32,223
490 DATA 126,223,35,126,167,196,24,0
500 DATA 121,223,62,35,35,241,201,2,4
510 DATA 8,3,37,33,35,36,0,0,0,0
```

KARATE TIL SVI 318/328

```
1 CLS
2 PRINT"^*^*^*^ KARATE ^*^*^*^"
5 INPUT"VILKEN SVARIGHETS GRAD";J
10 SCREEN 1,3
15 GOTO 2000
20 DATA &B00000011,&B10000000
30 DATA &B00000011,&B10000000
40 DATA &B00000011,&B10000000
50 DATA &B00000001,&B00000000
60 DATA &B00000111,&B11000000
70 DATA &B00000111,&B11000000
80 DATA &B00000111,&B11000000
90 DATA &B00000111,&B11000000
100 DATA&B00000111,&B11000000
110 DATA&B00000011,&B10000000
120 DATA&B00000011,&B10000000
130 DATA&B00000010,&B10000000
140 DATA&B00000010,&B10000000
150 DATA&B00000010,&B10000000
160 DATA&B00000010,&B10000000
170 DATA&B00000010,&B10000000
400 DATA &B00000000,&B00000000
410 DATA &B11100000,&B00011000
420 DATA &B11100000,&B00100000
430 DATA &B11111111,&B01000000
440 DATA &B00111100,&B10000000
450 DATA &B00111111,&B00000000
460 DATA &B00111111,&B00000000
470 DATA &B00100111,&B00000000
480 DATA &B00111011,&B00000000
490 DATA &B00000001,&B00000000
500 DATA &B00000001,&B00000000
510 DATA &B00000001,&B00000000
520 DATA &B00000001,&B00000000
530 DATA &B00000001,&B00000000
540 DATA &B00000001,&B00000000
545 DATA &B00000000,&B00000000
550 DATA &B00000001,&B11000000
560 DATA &B00000001,&B11000000
570 DATA &B00000001,&B11000000
580 DATA &E00000000,&B10000000
590 DATA &B00000001,&B11111110
600 DATA &B00000001,&B11000000
610 DATA &B00000001,&B11000000
620 DATA &B00000001,&B11111000
630 DATA &B00000001,&B11000000
640 DATA &B00000001,&B11000000
650 DATA &B00000001,&B11000000
660 DATA &B00000001,&B01000000
670 DATA &B00000001,&B00100000
680 DATA &B00000010,&B00010000
690 DATA &B00000100,&B00001000
700 DATA &B00001000,&B00001000
720 DATA &B00000000,&B00000000
730 DATA &B00000000,&B00000000
740 DATA &B00000000,&B00000000
750 DATA &B00000011,&B10000000
760 DATA &B00000011,&B10000000
770 DATA &B00000011,&B10000000
780 DATA &B0000C001,&B00000000
790 DATA &B00000111,&B11000000
800 DATA &B00000111,&B11000000
810 DATA &B00001011,&B10100000
820 DATA &B00010011,&B10010000
830 DATA &B00100011,&B10001000
840 DATA &B01000011,&B10000100
850 DATA &B10000011,&B10000010
860 DATA &B11111111,&B11111110
870 DATA &B00000000,&B00000000
880 DATA &B00000000,&B00000000
900 DATA &B00011000,&B00000111
910 DATA &B00000100,&B00000111
920 DATA &B00000010,&B11111111
930 DATA &B00000001,&B00111100
940 DATA &B00000000,&B11111100
950 DATA &B00000000,&B11111100
960 DATA &B00000000,&B11100100
970 DATA &B00000000,&B11011100
980 DATA &B00000000,&B10000000
990 DATA &B00000000,&B10000000
1000 DATA &B00000000,&B10000000
1010 DATA &B00000000,&B10000000
1020 DATA &B00000000,&B10000000
1030 DATA &B00000000,&B10000000
1035 DATA &B00000000,&B00000000
1050 DATA &B00000011,&B10000000
1060 DATA &B00000011,&B10000000
1070 DATA &B00000011,&B10000000
1080 DATA &B00000001,&B00000000
1090 DATA &B01111111,&B10000000
1100 DATA &B00000011,&B10000000
1110 DATA &B00000011,&B10000000
1120 DATA &B00011111,&B10000000
1130 DATA &B00000011,&B10000000
1140 DATA &B00000011,&B10000000
1150 DATA &B00000011,&B10000000
1160 DATA &B00000010,&B10000000
1170 DATA &B00000100,&B10000000
1180 DATA &B00001000,&B01000000
1190 DATA &B00010000,&B00100000
1195 DATA &B00010000,&B00010000
1196 DATA &B00000000,&B00000000
1198 DATA &B00000000,&B00000000
1199 DATA &B00000000,&B00000000
1200 DATA &B00000001,&B11000000
1201 DATA &B00000001,&B11000000
1202 DATA &B00000001,&B11000000
1203 DATA &B00000000,&B10000000
1204 DATA &B00000011,&B11111110
1205 DATA &B00000101,&B11000000
1206 DATA &B00000011,&B11000000
1207 DATA &B00000001,&B10000000
1208 DATA &B00000001,&B11000000
1209 DATA &B00000001,&B00100000
1210 DATA &B00000001,&B00100000
1211 DATA &B00000001,&B00100000
1212 DATA &B00001111,&B00100000
1213 DATA &B00000000,&B00000000
1214 DATA &B00000000,&B00000000
1215 DATA &B00000000,&B00000000
1216 DATA &B00000011,&B10000000
1218 DATA &B00000011,&B10000000
1219 DATA &B00000011,&B10000000
1220 DATA &B00000001,&B00000000
1221 DATA &B01111111,&B11000000
1222 DATA &B00000011,&B10100000
1223 DATA &B00000011,&B11000000
1224 DATA &B00000001,&B10000000
1225 DATA &B00000011,&B10000000
1226 DATA &B00000100,&B10000000
1227 DATA &B00000100,&B10000000
1228 DATA &B00000100,&B10000000
1229 DATA &B00000100,&B11110000
1230 DATA &B00000000,&B00000000
1244 FORS=0TO7
1245 S$=""
1246 T$=""
1247 FORA=1TO16
1248 READ X,Y
1249 S$=S$+CHR$(X)
1250 T$=T$+CHR$(Y)
```

```
1251 NEXT A
1260 SPRITE$(S)=S$+T$
1265 NEXT S
1280 Q=18
1285 U=1:F=1
1290 W=115
1295 S=0:Z=0
1296 P=192:O=115
1297 PUT SPRITE 3,(110,30),15,3
1300 PUT SPRITE 1,(Q,W),1,S
1301 PUT SPRITE 2,(P,O),9,Z
1304 IF Q>183 THEN Q=183
1305 IF P<40 THEN P=40
1306 IF P>230 THEN P=230
1307 IF Q<5 THEN Q=5
1308 IF P<Q+14 AND P>Q THEN 3000 EL
SE 1309
1309 IF Q>P-15 AND Q<P THEN 2200 EL
SE 1310
1310 ON STICK(1) GOSUB 1320,1330,13
40,1350,1360,1370,1380,1390
1311 ON STICK(2) GOSUB 2500,2510,25
20,2530,2540,2550,2560,2570
1312 IF Q<5 THEN Q=5
1313 IF P>250 THEN P=250
1315 GOTO 1300
1320 IF STRIG (1) THEN S=2 ELSE S=0
:RETURN
1330 RETURN
1340 Q=Q+J:IF STRIG(1) THEN S=6 ELS
E S=0:RETURN
1350 RETURN
1360 IF STRIG(1) THEN S=1ELSE S=0:R
ETURN
1370 RETURN
1380 Q=Q-J:RETURN
1390 RETURN
1400 RETURN
2000 COLOR 1,5,5
2010 SCREEN 1
2020 LINE(0,145)-(255,145),2
2021 LOCATE 0,110
2022 PRINT"┬┬┬┬┬┬┬┐"
2023 PRINT"┼┼┼┼┼┼┼┤"
2024 PRINT"┼┼┼┼┼┼┼┤"
2025 PRINT"┼┼┼┼┼┼┼┤"
2026 COLOR 9,1:LOCATE 204,110
2027 PRINT"┌┬┬┬┬┬┬┬"
2028 PRINT"
         ├┼┼┼┼┼┼┼"
2029 PRINT"
         ├┼┼┼┼┼┼┼"
2030 PRINT"
         ├┼┼┼┼┼┼┼"
2039 PAINT(150,160),2
2040 CIRCLE(100,5),20,6
2050 PAINT(100,5),6
2060 COLOR 1,3:LOCATE 90,150
2070 PRINT"| ╱┌┐┌┐┌┬┌─"
2080 PRINT"                      |╱ H H H
|├"
2090 PRINT"    ╱‾╲              |╲ |||╲||
||       ╱‾╲"
2100 PRINT"   ╱   ╲              | ╲─────
─└─      ╱   ╲"
2110 PRINT"                     =========
===="
2115 PLAY"L3003EBGE04C03GE04DDD03DG
GF#F#"
2116 PLAY"L3003EBGE04C03GE03DDD05DG
GF#F#03EEE"
2117 LINE(100,50)-(150,70),4,BF
2120 COLOR 3,5,1
2130 GOTO 20
2200 IF S=1 OR S=2 OR S=6 THEN 2205
 ELSE 1310
2205 COLOR 3,5,1:LOCATE 48,50
2207 SOUND 7,7:SOUND 8,16:SOUND 12,
30:SOUND 11,0:SOUND 13,1:FORD=1TO50
0:NEXTD:SOUND 8,0:SOUND 7,8:SOUND 8
,16
2210 PRINT"| | | ┬"
2220 PRINT"          ├─┤ | | "
2230 PRINT"          | | | | "
2240 PRINT"          ▒▒▒▒▒▒▒▒"
2242 LOCATE P,O-20:PRINT"OH SHIT!"
2244 FORA=1TO500:NEXTA
2245 COLOR 5,5:LOCATE 48,50
2246 PRINT"| | | ┬"
2247 PRINT"          ├─┤ | | "
2248 PRINT"          | | | | "
2249 PRINT"          ▒▒▒▒▒▒▒▒"
2250 LOCATE P,O-20:PRINT"OH SHIT!"
2251 Q=70:P=150:S=0
2252 U=U+1:GOTO 4025
2500 IF STRIG (2) THEN Z=5 ELSE Z=0
:RETURN
2510 RETURN
2520 P=P+J:RETURN
2530 RETURN
2540 IF STRIG(2) THEN Z=4 ELSE Z=0:
RETURN
2550 RETURN
2560 P=P-J:IF STRIG(2) THEN Z=7 ELS
E Z=0:RETURN
2570 RETURN
3000 IF Z=4 OR Z=5 OR Z=7 THEN 3010
 ELSE 1309
3010 COLOR 3,5,1:LOCATE 150,50
3015 SOUND 7,7:SOUND 8,16:SOUND 12,
30:SOUND 11,0:SOUND 13,1:FORD=1TO50
0:NEXT D:SOUND 8,0:SOUND 7,8:SOUND
8,16
3020 PRINT"| | | ┬"
3030 PRINT"
 ├─┤ | | "
3040 PRINT"
 | | | | "
3050 PRINT"
 ▒▒▒▒▒▒▒▒"
3055 LOCATE Q,W-20:PRINT"OH NO!"
3060 FORA=1TO500:NEXTA
3070 COLOR 5,5:LOCATE 150,50
3080 PRINT"| | | ┬"
3090 PRINT"
 ├─┤ | | "
3095 PRINT"
 | | | | "
4000 PRINT"
 ▒▒▒▒▒▒▒▒"
4005 LOCATE Q,W-20:PRINT"OH NO!"
4010 P=150:Q=70:Z=0
4020 F=F+1:GOTO 5500
4025 COLOR 3,5:LOCATE 10,10:PRINT"S
CORE:";U
4030 FORA=1TO500:NEXT A:IF U=10 THE
N 4035 ELSE 4031
4031 FORA=1TO500:NEXT A:IF U=20 THE
N GOTO 6000
4032 COLOR 5,5:LOCATE 12,10:PRINT"
█████"
4033 GOTO 1310
4035 COLOR 15,5:LOCATE 102,10
4040 PRINT"┌──────────┐"
4050 PRINT"                    | VERY G
OOD |"
```

```
4060 PRINT"                    |
    |"
4070 FORA=1TO1000:NEXT A
4080 COLOR 5,5:LOCATE 102,10
4090 PRINT"┌──────────┐"
5000 PRINT"                    | VERY G
OOD |"
5010 PRINT"                    |
    |"
5020 CIRCLE(100,5),20,6:PAINT(100,5
),6:GOTO 1310
5500 COLOR 3,5:LOCATE 200,10:PRINT"
SCORE:";F
5505 FORA=1TO500:NEXT A:IF F=10 THE
N GOTO 5535 ELSE 5510
5510 FORA=1TO500:NEXT A:IF F=20 THE
N GOTO 7000
5520 COLOR 5,5:LOCATE 202,10:PRINT"
    [redacted]"
5530 GOTO 1310
5535 COLOR 15,5:LOCATE 102,10
5540 PRINT"┌──────────┐"
5560 PRINT"                    | VERY G
OOD |"
5570 PRINT"                    |
    |"
5580 FORA=1TO1000:NEXT A
5590 COLOR 5,5:LOCATE 102,10
5600 PRINT"┌──────────┐"
5610 PRINT"                    | VERY G
OOD |"
5620 PRINT"                    |
    |"
5630 CIRCLE(100,5),20,6:PAINT(100,5
),6:GOTO 1310
6000 SCREEN 1,3:LINE(0,0)-(255,192)
,15,BF
6010 CIRCLE(128,96),90,9
6020 PAINT(128,96),9
6040 COLOR 1,15,15:LOCATE 50,80
6050 PRINT"BLACK WARRIOR IS THE WIN
NER"
6060 PRINT"          THANKS FOR A GO
OD FIGHT"
6065 FORA=1TO2000:NEXTA
6070 END
7000 SCREEN 1,3:LINE(0,0)-(255,192)
,15,BF
7010 CIRCLE(128,96),90,9
7020 PAINT(128,96),9
7040 LOCATE 50,80
7050 PRINT"RED WARRIOR IS THE WINNE
R"
7055 PRINT"          THANKS FOR A GO
OD FIGHT"
7060 FORA=1TO2000:NEXTA
7070 END
```

Med venlig Hilsen
Magnus Buhre

INDMELDELSESBLANKET

FORNAVN: ______________________________

EFTERNAVN: ______________________________

ADRESSE: ______________________________

TLFNR.: ______________________________

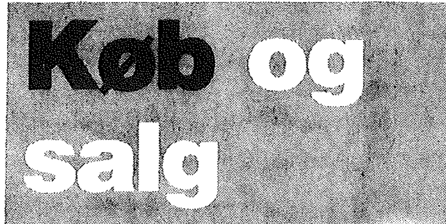
POSTNR. +BY: ______________________________

ALDER: ______________________________

MASKINE: ______________________________

DIV. UDSTYR: ______________________________

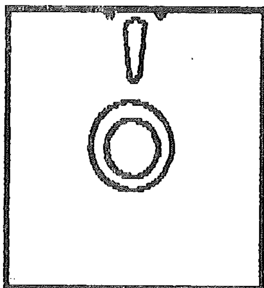
INTERESSE OMRADE: ______________________________



KØBES

MSX diskettestation (3.5" eller 5.25") med controler købes.

Henv. Kim Andersen
02 94 26 74



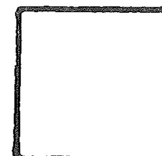
Kan du skaffe klubben nye medlemmer har du muligheden for få bånd eller disetter i præmie.
Du må ikke selv være "skaffet" af andre!
Du skal selv være medlem i 1989 !
Man kan altså ikke melde hinanden ind i klubben og derved opnå dobbelt præmie.
Ved et nyt medlem forstås en der ikke var medlem i 1988 !

Præmie :

Pr. indmeldt medlem kan du få
2 bånd
eller
5 stk. 5 1/4 '' disk
eller
2 stk. 3 1/2 '' disk

Melder du et medlem ind skal han/hen selv skrive dit navn og medlemnr. på bagsiden af Girokorte samt hvad du ønsker som præmie.

KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP



MSX BRUGER-KLUBBEN
KASSERER PREBEN LUND
GALGEBAKKEN NEDER 2-3B
2620 ALBERTSLUND




SVI BIB

SVI BIB er KUN til SVI 318-328

SVIBIBS's POSTGIRONR. er:
4077385

Bestilling af programmer gøres på postgiro hvorpå der skrives hvilke programmer du ønsker,samt om du vil have disk eller bånd.
Husk også at skrive din egen adr. tydeligt helst **BLOKBOGSTAVER.**

PRISER

Bånd	-	25kr.
5.25" Disk	-	25kr.
Pris pr. program	-	10kr.
Pris pr. programpakke	-	15kr.
Udlistning af program	-	5kr.

Programpakkerne er lavet på den måde at man loader alle programmerne ind på en gang , og kan så vælge ved hjælp af en meny hvilket program man ønsker at køre derefter kan man stoppe og vende tilbage til menyen ovs.ovs.

HUSK når I bestiller programpakker at skrive hvilken katarogi I ønsker det pågældende nr. fra.

Programmerne bestilles hos:

Per Underlien
Rytterhusene 44
2620 Albertslund
GIRONR. 4077385

Programmer der ønskes optaget i SVIBIB sker på samme adr.
Det er jo sådan at vi NU GIVER
2 bånd eller
5 5.25" disk
For de programmer der optages i SVI BIB.

Hilsen Per

Liste af programtitler i SVIBIB :

Nyttige programmer :

Database , Editor , Hushold ,Matte, Mc scroll , Ramdis , Rentereg , Rulleteks , Screen Rig , Sdump.mc, Sorter , Sprite ed , Sprog , Tegner , Talkback , Taperut , Tips Kalender , Supertegn , Periode

Spil :

Aladdin , Bovla, Bjerg, Darts, Death, Diktator, Frans, Hell, Kalaha, Skyd nu, Oil

Spil pakker :

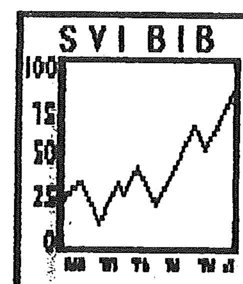
Nr. 1 : Male , Pyramide , Labyrant
Nr. 2 : Pusle , Attack , Roulette
Nr. 3 : Flight , Sur , War
Nr. 4 : Mons , Hopper , Star
Nr. 5 : Moon , Myre , Slange
Nr. 6 : Robot , Pengo , Snowball
Nr. 7 : Danmark , Frogger
Nr. 8 : Skyttepro , Lemona

Musik pakker :

Nr. 1 : Rock musik 9 numre
Nr. 2 : Rock musik 9 numre
Nr. 3 : Musik 9 numre
Nr. 4 : Lyd Kim , Sound Kim , Orgel

Andet pakker :

Nr. 1 : Rio , Lys , Ghost , Børge
Symtri , Grafik



Kun til disk :

Budgetpro. , Dbase 1 no.1 , Grafen
Tegnepro. , Telefonpro.

DIRECTORY FOR TAPEFANS.

Her er et "lille" program der ved ind- og afspilning af basicprogrammer kan holde lidt styr på indholdet. Programmet er aldeles ikke selvforklarende. Men som en hjælp kan man bare trykke på funktionstast F1, så får du brugsanvisningen som så forhåbentligt gi'r den hjælp der skal til.

Programmet poker informationen ind på de linier der ser "tomme" ud, og ved udlæsning peeker den så ud igen til skærm. Hvis du er lidt doven kan du undlade hjælpeteksten linierne fra 1000 til 1080.

Det var forudset, at programmet også skulle understøtte maskinprogrammer, men så fik jeg en disk og fik ikke gjort mere ved det. Men den kvikke prøver sikkert med en udvidelse.

Cementkogeren.

```
10 ' (C) PETER WHITE 860323
20 DATA        ,        ,        ,
   ,        ,        ,        ,
,        ,        ,        ,        ,
     ,        ,        ,        ,
   ,        ,        ,        ,
30 DATA          ,          ,
,          ,          ,          ,
     ,          ,          ,          ,
          ,          ,          ,
     ,          ,          ,          ,
          ,          ,          ,
50 COLOR15,1,12:KEYON
60 DIM F$(20),C$(20):SCREENO:WIDTH 39
70 Z=6:P=32806!:RESTORE
80 FORX=0TO20:READF$(X):NEXT
90 FORX=0TO20:READC$(X):NEXT:LOCATE0,0
100 PRINT"TAPE ";F$(0):LOCATE14,0:PRINT
"NR  PROGRAM   TÆLLER",,,"DIRECTORY",,,
,"SIDE ";C$(0)
110 KEY1,"HJÆLP":KEY2,"INIT":KEY3,"SAVE
":KEY4,"?LOAD":KEY5,"INSERT"
120 FORX=1TO20:LOCATE 14,X:PRINTUSING"#
#";X
130 LOCATE19,X:PRINTF$(X)
140 LOCATE28,X:PRINTC$(X)
150 NEXTX
160 FORX=1TO5:KEY(X)ON:NEXTX
170 GOSUB420
180 ON KEY GOSUB 1000,440,620,710,200
190 GOTO180
200 LOCATE0,21:INPUT "NR";X:Z=8:P=32795
!
210 IFX>20 THEN GOTO 200
220 IF X=0 THEN GOTO370
230 LOCATE17,X:PRINT"*"
240 GOSUB 420:INPUT"PROGRAM";F$(X)
250 IF LEN (F$(X))>6 THEN 230
260 LOCATE19,X:PRINTF$(X):GOSUB560
270 LOCATE26,X:PRINT"*"
280 GOSUB420:PRINT"?LOAD":L$=INPUT$(1)
290 IF L$=" " OR L$="B" OR L$="C" OR L$
="b" OR L$="c" THEN GOTO 300 ELSE 280
300 LOCATE 26,X:PRINTL$
310 POKE 32812!+(X*9),ASC(L$)
320 GOSUB 420:INPUT"TÆLLER";F$(X)
330 LOCATE 28,X:PRINTF$(X)
340 GOSUB550
350 LOCATE17,X:PRINT" "
360 GOSUB 420:RETURN 70
370 GOSUB 420:INPUT"TAPE ";F$(0)
380 LOCATE 5,0:PRINTF$(0):GOSUB 560
390 GOSUB 420:PRINT"SIDE ";:S$=INPUT$(1
)
400 F$(0)=S$:GOSUB 550
410 LOCATE 5,4:PRINTC$(0):RETURN 70
420 LOCATE 0,21:PRINT SPC(38)
430 LOCATE0,21:GOSUB690:RETURN
440 BEEP
450 FORV=32805!TO32985!STEP9
460 FORW=0TO7
470 POKE(W+V),32
480 NEXTW:NEXTV
490 FORV=32999!TO33219!STEP11
500 FORW=0TO9
510 POKE(W+V),32
520 NEXTW:NEXTV
530 CLS
540 RETURN70
550 Z=10:P=32987!
560 IF LEN(F$(X)) >Z THEN F$(X)=LEFT$(F
$(X),Z)
570 FOR Y=1 TO Z
580 IF LEN(F$(X))=Z THEN GOTO 600
590 F$(X)=F$(X)+" ":GOTO580
600 S=ASC(MID$(F$(X),Y,1))
610 POKE P+((Z+1)*(X+1))+Y,S:NEXTY:RETU
RN
620 LOCATE0,21:PRINT"SPOL TILBAGE TIL "
;C$(1);" TRYK ENTER";
630 T$=INPUT$(1):GOSUB420
640 LOCATE0,21:PRINT"SAVING: ";F$(1)
650 GOSUB690
660 CSAVE "DIR #1",2:GOSUB420:FORA=1TO5
670 PRINT"READY":GOSUB700:GOSUB420
680 GOSUB700:NEXTA:RETURN
```

```
690 BEEP:RETURN
700 FORT=1TO99:NEXTT:RETURN
710 INPUT"NR";X:IFX=0THEN710
720 L$=LEFT$(F$(X),6)
730 CLS:PRINT"DU SKAL SPOLE HEN TIL: ";
C$(X),,,,,"FOR AT LOADE: ";L$,,,,,:GOSU
B790
740 PRINT"NAR DU HAR GJORT DET, TRYK EN
 TAST":T$=INPUT$(1):PRINT,,,"OG START B
ANDOPTAGEREN",,,,:GOSUB790
750 PRINT"DIT PROGRAM BLIVER LOADET"
760 S$=RIGHT$(F$(X),1)+"LOAD"+CHR$(34)+
L$+CHR$(34)+CHR$(13)+"RUN"+CHR$(13)
770 CLOAD
780 RETURN
790 FORT=0TO555:NEXTT:RETURN
1000 'HJÆLP
1010 CLS:PRINT"DU HAR BENYTTET HJÆLP !"
,,,"PROGRAMMET BENYTTES TIL AT HOLDE ST
YR  PÅ DINE BANDEDE PROGRAMMER.",,,"HVI
S DU TRYKKER PÅ F2 (INIT) SLETTES   ALL
E DINE DATA, BRUGES NAR DU TAGER ET NYT
 BAND I BRUG."
1020 PRINT,,"F3 (SAVE) GEMMER DIN INDHO
LDSFORTEGNEL-SE, HUSK DET, HVIS DU HAR
LAVET RETTEL-SER, OG INDEN DU LOADER ET
 PROGRAM.",,,"F4 (?LOAD) LOADER DET DU
HAR VALGT."
1030 PRINT,,"F5 (INSERT), HER INDTASTER
 DU NYE DATA I DIN INDHOLDSFORTEGNELSE.
 0=TAPE-NR.",,,"NAR DU HAR LÆST DENNE S
IDE FÆRDIG, TRYKEN TAST."
1040 T$=INPUT$(1):CLS
1050 PRINT"HVER GANG DU BRUGER ?LOAD EL
LER INSERT,VIL PROGRAMMET SPØRGE OM ET
NR, OG DET ER DIN INDGANG TIL DET AKTUE
LLE PROGRAM",,"DU MÅ SOM PROGRAMNAVN BR
UGE OP TIL SEKSTEGN OG VED TÆLLERPOSITI
ONEN OP TIL 10 TEGN, OGSÅ ALFANUMERISKE
."
1060 PRINT,,"MEN VÆR SÅ FORNUFTIG AT BR
UGE NAVNE,DERSIGER NOGET OM INDHOLDET.
T-LLERPOSITI-ONEN KAN SÅ BRUGES TIL AT
FORTÆLLE OM  LÆNGDEN AF PROGRAMMET.",,,
"F.EKS.: DIR #1, FRA 10-20, SÅ RISIKERE
RDU IKKE AT OVERSPILLE ET EVENTUELT AN-
 DET PROGRAM."
1070 PRINT,,"LAV OGSÅ ET PÆNT MELLEMRUM
 MELLEM PRO- GRAMMERNE, SÅ DU HAR PLADS
 NOK VED EN  UDVIDELSE AF PROGRAMMERNE.
",,,"RETUR TIL PROGRAM, TRYK EN TAST."
1080 T$=INPUT$(1):CLEAR:GOTO 60
1090 FOR M=32768! TO 33500!
1100 PRINTM;"=";PEEK(M);"=";HEX$(PEEK(M
));"=";CHR$(PEEK(M))
1110 FOR T=0TO25:NEXTT
1120 NEXT M
```

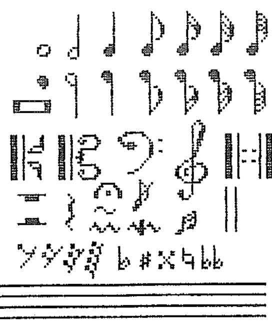
DX7 RANDOM PATCH GENERATOR FOR MSX 2 & SFG UNIT.

```
10 'DX7 Random patch generator for
20 'MSX2 & SFG unit
30 '  (C) by Gabriel Landini 1988
40 '
50 'Fodnote af Johnny B.
60 '
70 'Denne rutine generere en lyd
80 'til din tilsluttede DX7 og kan
90 'regenerere udfra de samme
100 'værdier
110 '
120 'Hvis nogen kan opstille
130 'programmet Grafisk så ville
140 'det jo være en god ide
150 'Det skulle være således at
160 'lydkurven bliver Grafisk
170 'tegnet og man kunne gå ind
180 'med Cursor eller Mus og
190 'ændre kurven således at
200 'værdierne så bliver genere-
210 'ret udfra den nye kurve
220 '
230 'Kontakt mig hvis du har en
240 'eller et program der evt
250 'omskrives til formålet.
260 'Hej fra Johnny B
270 '
280 POKE &H9FF4,PEEK(&HFC4A):POKE &
H9FF5,PEEK(&HFC4B)
290 CLEAR 40,&H9BFF:BLOAD "d":DEFIN
T A-Z
300 FOR A=&H9FF6 TO &H9FFF:POKEA,0:
NEXT:POKE &H9FF1,0
310 DEFUSR=&H9C00
320 POKE &H9FFC,2:A=USR(0)
330 POKE &H9FFC,1
340 CLS:PRINT SPC(30)"EG Rates":PRI
NT:PRINT:PRINT:FOR X=1 TO 4
350 PRINT"     Rate ";X;" upper limi
t (0-99):";:INPUT R1(X)
360 IF R1(X)>99 OR R1(X)<0 THEN GOS
UB 1560:GOTO 350
370 PRINT"     Rate ";X;" lower limi
t (0-99):";:INPUT R2(X)
380 IF R2(X)>R1(X) OR R2(X)<0 THEN
GOSUB 1560:GOTO 370
390 PRINT:NEXT
400 CLS:PRINT SPC(30)"EG Levels":PR
INT:PRINT:PRINT:FOR X=1 TO 4
410 PRINT"     Level ";X;" upper lim
it (0-99):";:INPUT L1(X)
420 IF L1(X)>99 OR L1(X) <0 THEN GO
SUB 1560:GOTO 410
```

```
430 PRINT"    Level ";X;" lower lim
it (0-99):";:INPUT L2(X)
440 IF L2(X)>L1(X) OR L2(X) <0 THEN
 GOSUB 1560:GOTO 430
450 PRINT:NEXT
460 CLS:PRINT SPC(22)"Operators lev
el & frequency":PRINT:PRINT:PRINT
470 PRINT "    Operators output upp
er limit (0-99):";:INPUT Q1
480 IF Q1>99 OR Q1<0 THEN GOSUB 156
0:GOTO 470
490 PRINT "    Operators output low
er limit (0-99):";:INPUT Q2:
500 IF Q2 >Q1 OR Q2<0 THEN GOSUB 15
60:GOTO  490
510 PRINT:PRINT "    Coarse frequen
cy ratio upper limit (0-31):";:INPU
T F1
520 IF F1 >31 OR F1<0 THEN GOSUB 15
60:GOTO 510
530 PRINT"    Coarse frequency rati
o lower limit (0-31):";: INPUT F2
540 IF F2>F1 OR F2< 0 THEN GOSUB 15
60:GOTO 530
550 PRINT:PRINT"    Fine frequency
ratio upper limit (0-99):";:INPUT F
3
560 IF F3>99 OR F3<0 THEN GOSUB1560
:GOTO 550
570 PRINT"    Fine frequency ratio
lower limit (0-99):";:INPUT F4
580 IF F4>F3 OR F4 <0 THEN GOSUB156
0:GOTO 570
590 FOR X=1 TO 4
600 RU(X)=R1(X)+1:LU(X)=L1(X)+1
610 RL(X)=R2(X):LL(X)=L2(X)
620 NEXT
630 QU=Q1+1:QL=Q2:FU=F1+1:FL=F2:FA=
F3+1:FB=F4
640 FOR X=1 TO 4
650 RU(X)=RU(X)-RL(X)
660 LU(X)=LU(X)-LL(X)
670 NEXT
680 X=RND(-TIME)
690 QU=QU-QL:FU=FU-FL:FA=FA-FB
700 FOR X=1 TO 6:FOR Y=1 TO 4
710 R(X,Y)=((RND(1)*RU(Y))+RL(Y))
720 L(X,Y)=((RND(1)*LU(Y))+LL(Y))
730 NEXT
740 Q(X)=((RND(1)*QU)+QL)
750 F(X)=(RND(1)*FU)+FL
760 FF(X)=(RND(1)*FA)+FB
770 NEXT
780 AL=INT(RND(1)*32)+1
790 CLS:PRINT"  DX 7   Random patch
 generator"
```

```
800 LOCATE2,2:PRINT"Algorithm:";AL;
" (not transmitted)
810 LOCATE 0,4:PRINT "  EG rate (R1
-R4)":PRINT
820 FOR X=1 TO 6
830 PRINT "Op.";X;":";
840 FOR Y=1 TO 4
850 PRINT USING" ##" ;R(X,Y);
860 NEXT:PRINT
870 NEXT
880 ZZ=24:Z=4
890 LOCATEZZ,Z:PRINT" EG level  (L1
-L4)"
900 FOR X=1 TO 6
910 LOCATEZZ,Z+2:PRINT"Op.";X;":";
920 Z=Z+1
930 FOR Y=1 TO 4
940 PRINT USING" ##";L(X,Y);
950 NEXT:PRINT
960 NEXT
970 ZZ=ZZ+24:Z=4
980 LOCATEZZ-1,Z:PRINT"Output level
":Z=Z+2
990 FORX=1 TO 6
1000 LOCATEZZ,Z:PRINT "Op.";X;":";Q
(X)
1010 Z=Z+1:NEXT
1020 ZZ=ZZ+15:Z=4
1030 LOCATEZZ,Z:PRINT" Freq. ratio"
:Z=Z+2
1040 FOR X=1 TO 6:KF(X)=F(X)
1050 'DX panel system
1060 IF F(X)=0 THEN FG!(X)=(FF!(X)/
2)+50:KF(X)=1:ELSE FG!(X)=FF(X)
1070 TF!(X)=(FG!(X)/100)*KF(X)+F(X)
1080 NC(X)=INT(TF!(X)):NF!(X)=(TF!(
X))-NC(X)
1090 LOCATEZZ,Z:PRINT "Op.";X;":";U
SING" ##";NC(X);:PRINTUSING".##";NF
!(X)
1100 Z=Z+1:NEXT
1110 LOCATE 0,20:PRINT"
***  BUSY TRANSMITTING EXCLUSIVE MI
DI MESSAGE   ***"
1120 C=0:FOR X=6 TO 1 STEP-1:FOR Y=
1 TO 4
1130 POKE &HA020,&HF0
1140 POKE &HA021,&H43
1150 POKE &HA022,&H10
1160 POKE &HA023,&H0
1170 POKE &HA024,C
1180 POKE &HA025,R(X,Y)
1190 POKE &HA026,&HF7:A=USR(0)
1200 C=C+1:IF C=4 THEN C=21
1210 IF C=25 THEN C=42
1220 IF C=46 THEN C=63
```

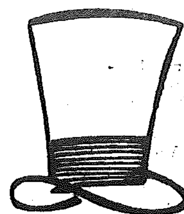
```
1230 IF C=67 THEN C=84
1240 IF C=88 THEN C=105
1250 NEXT:NEXT
1260 C=4:FOR X=6 TO 1 STEP-1:FOR Y=
1 TO 4
1270 POKE &HA024,C
1280 POKE &HA025,L(X,Y)
1290 A=USR(0)
1300 C=C+1:IF C=8 THEN C=25
1310 IF C=29 THEN C=46
1320 IF C=50 THEN C=67
1330 IF C=71 THEN C=88
1340 IF C=92 THEN C=109
1350 NEXT:NEXT
1360 C=16:FOR X=6 TO 1 STEP-1
1370 POKE &HA024,C
1380 POKE &HA025,Q(X)
1390 A=USR(0)
1400 C=C+21:NEXT
1410 C=18:FOR X=6 TO 1 STEP-1
1420 POKE &HA024,C
1430 POKE &HA025,F(X)
1440 A=USR(0)
1450 C=C+21:NEXT
1460 C=19:FOR X=6 TO 1 STEP-1
1470 POKE &HA024,C
1480 POKE &HA025,FF(X)
1490 A=USR(0)
1500 C=C+21:NEXT
1510 LOCATE 7,20:PRINT"Would you li
ke to generate a new set of values
   (y/n) ? ";:A$=INPUT$(1)
1520 PRINTA$:IF A$="n" OR A$="N" TH
EN END
1530 PRINT "        Would you like t
o use the same limits (y/n) ? ";:A$
=INPUT$(1)
1540 PRINT A$:IF A$="y" OR A$="Y" O
R A$=CHR$(13) THEN GOTO 590
1550 GOTO 340
1560 PRINT:PRINT"That is not a vali
d response."
1570 PRINT"Re-enter, please."
1580 RETURN
```



```
10 DATA F3,CD,38,1,32,FF,9F,5F,E,0,
6,0,21,C5,FC,9,56,21,C1,FC,9,6,4,7E
,28,12,6,0,79,F,F,AB,E6,C0,AB,D3,A8
,7A,E6,FC,B0,32,FF,FF,7B,E6,FC,B1,D
3,A8,C5,D5,CD,60,9C,D1,C1,28,67,4,7
8,FE,4,38,DB
20 DATA 20,10,79,F,F,AB,E6,C0,AB,D3
,A8,7A,32,FF,FF,7B,D3,A8,C,79,FE,4,
38,B1,3A,FF,9F,D3,A8,18,16,21,70,9C
,11,80,0,6,6,1A,BE,C0,23,13,10,F9,C
9,4D,43,48,46,4D,30,CD,D2,0,CD,C0,0
,21,8E,9C,6,14
30 DATA 7E,CD,A2,0,23,10,F9,CD,9F,0
,C3,7C,9C,4E,6F,74,20,46,6F,75,6E,6
4,20,53,46,47,20,55,6E,69,74,20,21,
3E,0,32,FE,9F,3A,FC,9F,FE,0,CA,F2,9
C,FE,1,CA,30,9D,FE,2,CA,61,9D,3A,FF
,9F,D3,A8,C9,3E,0
40 DATA 32,F9,9F,3A,FA,9F,FE,13,C8,
CD,61,9D,3E,FF,32,F9,9F,C9,3E,15,32
,F6,3F,C9,3A,F6,3F,CB,4F,CA,D9,9C,3
A,F5,3F,C9,3A,F6,3F,CB,47,C0,C3,E5,
9C,32,F5,3F,C9,CD,BF,9C,3A,F9,9F,FE
,FF,C8,CD,D3,9C,21,20
50 DATA A0,22,F2,9F,CD,D9,9C,FE,F0,
C2,4,9D,77,23,CD,D9,9C,77,FE,F7,CA,
2A,9D,FE,FF,CA,2A,9D,FE,F8,CA,E,9D,
FE,FE,CA,E,9D,23,C3,E,9D,22,F0,9F,C
3,55,9D,CD,BF,9C,3A,F9,9F,FE,FF,C8,
CD,D3,9C,21,20,A0,22,F2
60 DATA 9F,CD,E5,9C,7E,CD,EE,9C,FE,
F7,CA,52,9D,23,C3,42,9D,C3,55,9D,3E
,FF,32,FB,9F,3A,FF,9F,D3,A8,FB,C9,3
A,FF,9F,D3,A8,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
70 DATA 0,0,0,0,0,0,0,0,0,0,3E,13,3
2,FA,9F,C9,53,45,4D,20,5C,5C,20,44,
4F,20,4E,4F,54,20,4D,41,4B,45,20,41
,20,43,4F,50,59,20,3A,20,5C,5C,43,4
F,50,59,21,26,3,2E,4,CD,C6,0,6,1B,2
1,94,9D,7E
80 DATA CD,A2,0,23,10,F9,6,5,21,AF,
9D,CD,9F,0,BE,C2,D9,9D,23,10,F6,C3,
8B,9D,C3,0,0,F3,CD,38,1,FF,FF,0,0,F
F,FF,0,0,FF,FF,0,0,FF,FF,0,0,FF,FF,
0,0,FF,FF,0,0,FF,FF,0,0,FF,FF,0,0
90 '
1040 'GEMMER MASKINKODEN SOM BINÆR
FIL
1050 BSAVE"D",&H9C00,&H9DFF
1060 'D er filen du skal bruge i
1070 'DX7 Random programmet for
1080 'at overføre via SFG modulet
1090 'Hej Johnny B.
```

LINKE PROGRAM

```
10 '********************************
20 '* YAMAHA MUSIC COMPOSER II     *
30 '* FILE PART LINKER  ver 2.3    *
40 '* for MSX2 BASIC               *
50 '* (c) 1988 by                  *
60 '* Gabriel Landini              *
70 '* JAPAN                        *
80 '********************************
81 '
82 'Med dette program kan du
83 'linke parter fra composer
84 'direkte i Basic
85 'Hej Johnny B
86 '
90 'AD=load address : cF$=filename
: EP=extract part No.
100 'L=length of source file : I=co
unter : X1$, X2$, X3$=data :PA=part
 No.
110 'PB()=part begin address : PE()
=part end address: TR=end trap
120 'TL=length of linked file: BD$=
length of data: BL$=low byte :BH$=h
igh byte
130 CLEAR 100,&HAFFF: TR=0:DEFINT A
-Z
140 DEF FNA$=HEX$(ASC(INPUT$(1,1))+
ASC(INPUT$(1,1))*256)
150 AD=&HB000
160 CLS
170 PRINT"             PART LINKER FOR
YAMAHA MUSIC COMPOSER II FILES
v.2.3"
180 FILES "*.cmp":PRINT:PRINT
190 INPUT"Extract part from file >.
CMP<:";SF$
200 CF$=SF$+".cmp"
210 INPUT"Extract part No. (1-8):";
EP
220 GOSUB 530 :' offset bload addre
ss & bload
230 PA=0
240 PB(1)=50+AD
250 FOR I=AD-1  TO AD+L-2 STEP 3
260 X1$=HEX$(PEEK(I)):X2$=HEX$(PEEK
(I+1)):X3$=HEX$(PEEK(I+2))
270 PRINT HEX$(I),X1$,X2$,X3$,I-AD+
1;
280 IF X1$="FF" AND X2$="0" AND X3$
="0" THEN PA=PA+1:PRINT "  End of p
art ";PA: PE(PA)=I+2:PB(PA+1)=I+3
290 PRINT
300 NEXT
310 FOR ZZ=1 TO 8
320 PRINT "Part";ZZ;"starts in ";HE
X$(PB(ZZ)-AD);" ends in ";HEX$(PE(Z
Z)-AD);" and is";PE(ZZ)-PB(ZZ)+1;"b
ytes long >";(PE(ZZ)-PB(ZZ)-2)/3;"<
 events"
330 NEXT
340 IF EP=0 OR TR=1 THEN END
350 BSAVE "temp1.bin",PB(EP),PE(EP)
360 PRINT
370 INPUT"Link to file >.CMP<:";LF$
380 CF$=LF$+".cmp"
390 PRINT
400 GOSUB 530
410 PRINT
420 EE=L
430 BLOAD "temp1.bin",EE-3-PB(EP)+A
D
440 PRINT "linked.cmp";",";0;",";PE
(EP)-PB(EP)+EE-2+7
450 TL=(PE(EP)-PB(EP)+EE-2)
460 BD$=RIGHT$("000"+HEX$(TL-33),4)
:BL$=RIGHT$(BD$,2)
470 POKE 48+AD,VAL("&h"+BL$)
480 BH$=LEFT$(BD$,2):POKE 49+AD,VAL
("&h"+BH$)
490 BSAVE "linked.cmp",AD,TL+AD-1
500 KILL "temp1.bin"
510 EP=8:TR=1:L=TL
520 GOTO 230
530 OPEN CF$ FOR INPUT AS #1
540 L=LOF(1):PRINT CF$+" is ";L;"by
tes long":L=L-7
550 B=ASC(INPUT$(1,1))
560 IF B<>254 THEN PRINT"Bad file m
ode":CLOSE:END
570 PRINT:RA$=FNA$
580 PRINT"Start:&h";RA$;
590 PRINT"  End:&h";FNA$;
600 PRINT"  Exec:&h";FNA$:PRINT
610 CLOSE
620 BLOAD CF$,(AD-VAL("&h"+RA$))
630 RETURN
```



3 PROGRAMMER FRA MATTHIAS SVERIGE

Det ser ud til at vore Svenske kolleger er de mest aktive idet jeg igen har fået noget tilsendt fra hinsidan denne gang fra Matthias Gustafsson Horred Sverige Det drejer sig om 3 programmer:
Dragon-demo (en lille drage-movie)
Iron Maiden demo (Hard Rock)
Scarborough Fair (3 stemmig lyd)
Jeg takker Matthias mange gange og en lille hilsen er også gået til Sverige, det kan han jo så fortælle sine kammarater om og i andre kan finde ud af det ved at sende mig noget som i har lavet på Los Computer

Fra Tonny Marcher i Esbjerg har jeg også fået tilsendt nogle programmer, det drejer sig om tekstbehandling samt database og lønprogrammer, disse prg kan du erhverve hos Muco-bib eller hos MSX-BIB det drejer sig om ialt 3 programmer til sædvanlig pris Også en stor tak til Tonny flot arbejde.
Nu til Matthias's 3 programmer

Hej fra Johnny B Musik
Lindevænget 237
8600 Silkeborg

```
1 ' EN JULKLAPP TILL SYNTAXEN IFRAN
2 ' GARF XXX alias MATTIAS GUSTAFSS
ON
3 ' HEJ TIL JOHNNY B.!!!
4 '
10 ON STOP GOSUB 1230:STOP ON
20 GOSUB 1150
30 COLOR 1,15,15
40 SCREEN 2,,0:X=0:Y=0
50 ' border
60 FOR A=1 TO 8:READ B:IF B=1 THEN
X=X+1:Y=Y+1:GOTO 90
70 LINE(X,Y)-(255-X,191-Y),4,B
80 X=X+1:Y=Y+1
90 NEXT
100 DATA 0,0,1,0,1,0,1,1
110 ' space + stars
120 LINE(X+1,Y+1)-(255-X-1,191-Y-1)
,1,BF
130 FOR Y=9 TO 182STEP8
140 FOR X=9 TO 246STEP8
150 B=INT(RND(1)*20)
160 IF B<2 THEN PSET(X,Y),15
170 NEXT X:NEXT Y
180 ' dragon
190 PRESET(170,80):FOR A=1 TO 64:RE
AD X,Y,C:LINE-STEP (X,Y),C:NEXTA
200 DATA -10,10,2,-5,10,2,0,10,2,5,
10,2,10,5,2,5,7,2,0,5,2,-5,2,2,-10,
0,2,-5,5,2,0,2,2,5,2,2,2,-2,2,2,0,2
,2,2,2,2,0,2,2,-2,2,10,-3,2,2,-5,2,
2,0,2,0,10,2,-2,2,2,0,2,2,2,2,2,2,0
,2,2,-2,2,0,-2,2
210 DATA 2,0,2,2,2,2,2,0,2,2,-2,2,0
,-5,2,-5,-10,2,0,-10,2,6,0,2,20,13,
2,2,0,2,2,-2,2,-5,-3,2,-30,-40,2,0,
-5,2,-1,-7,2,-5,-5,2,2,-2,2,3,-7,2,
5,-5,2,-2,0,2,-5,-5,2,-3,-7,2,-2,-2
,2,-2,0,2,-2,2,2,-2,0,2,-2,-2,2,-2,
0,2,-2,2,2
220 DATA -3,10,2,-10,3,2,-5,5,2,0,5
,2,5,5,2,10,0,2,10,-2,2,-3,5,2
230 PAINT(180,100),2
240 ' eyes
250 CIRCLE(170,57),4,15,,,2
260 CIRCLE(173,57),4,15,,,2
270 PAINT(170,57),15
280 PAINT(173,57),15
290 ' mouth
300 LINE(170,75)-(178,69),6
310 LINE(178,69)-(174,75),6
320 LINE(174,75)-(170,75),6
330 PAINT(173,74),6
340 ' teeth
350 LINE(153,79)-(156,82),15,BF
360 LINE(158,79)-(161,82),15,BF
370 ' nose
380 CIRCLE(154,71),1,1:PAINT(154,71
),1
390 CIRCLE(159,71),1,1:PAINT(159,71
),1
400 ' wings
410 LINE(180,90)-(170,110),7
420 LINE(170,110)-(175,115),7
430 LINE(175,115)-(185,120),7
440 LINE(185,120)-(195,123),7
450 LINE(195,123)-(179,91),7
460 PAINT(180,100),7
470 LINE(177,102)-(183,108),1
480 LINE(177,110)-(183,116),1
490 LINE(183,116)-(188,117),1
500 ' dot on stomack
510 CIRCLE(160,105),1,1
520 ' legs
```

```
530 LINE(184,140)-(184,135),1
540 LINE(184,135)-(180,132),1
550 LINE(180,132)-(170,130),1
560 ' sprites
570 RESTORE 800
580 FOR B=1 TO 8
590 READ A$
600 B$=CHR$(VAL("&b"+A$))
610 C$=C$+B$
620 NEXT B
630 SPRITE$(2)=C$
640 RESTORE 710
650 FOR E=1 TO 8
660 READ R$
670 U$=CHR$(VAL("&b"+R$))
680 O$=O$+U$
690 NEXT E
700 SPRITE$(1)=O$
710 DATA 00000000
720 DATA 00101000
730 DATA 01101100
740 DATA 01101100
750 DATA 01101100
760 DATA 00101000
770 DATA 00000000
780 DATA 00000000
790 '
800 DATA 00000000
810 DATA 11100111
820 DATA 11100111
830 DATA 00011000
840 DATA 00111100
850 DATA 00011000
860 DATA 00100100
870 DATA 01011010
880 PUT SPRITE 1,(168,53),1
890 PUT SPRITE 2,(50,50),11
900 X=50:Y=50:XX=168:YY=53:Q=168:U=
53
910 A=STICK(D)
920 IF A=1 THEN Y=Y-4
930 IF A=2 THEN Y=Y-4:X=X+4
940 IF A=3 THEN X=X+4
950 IF A=4 THEN X=X+4:Y=Y+4
960 IF A=5 THEN Y=Y+4
970 IF A=6 THEN X=X-4:Y=Y+4
980 IF A=7 THEN X=X-4
990 IF A=8 THEN X=X-4:Y=Y-4
1000 IF X>XX THEN XX=XX+1
1010 IF X<XX THEN XX=XX-1
1020 IF Y>YY THEN YY=YY+1
1030 IF Y<YY THEN YY=YY-1
1040 IF XX>Q+2 THEN XX=XX-1
1050 IF XX<Q-2 THEN XX=XX+1
1060 IF YY>U+2 THEN YY=YY-1
1070 IF YY<U-2 THEN YY=YY+1
1080 PUT SPRITE1,(XX,YY),1
1090 PUT SPRITE2,(X,Y),11
1100 IF X<9 THEN X=9
1110 IF X>238 THEN X=238
1120 IF Y>174 THEN Y=174
1130 IF Y<9 THEN Y=9
1140 GOTO 910
1150 SCREEN 0:COLOR 15,4,4
1160 CLS:KEY OFF
1170 PRINT"     T I L L   S Y N T A
 X"
1180 PRINT:PRINT:PRINT"      Från Ma
ttias.G":PRINT:PRINT:PRINT
1190 INPUT"Joystick/Keyboard (J/K)"
;D$
1200 IF D$="J"ORD$="j"THEN D=1
1210 IF D$="K"ORD$="k"THEN D=0
1220 RETURN
1230 SCREEN 0:COLOR 15,4,4
1240 PRINT"Do you want to stop..(Y/
N)
1250 A$=INKEY$:IFA$=""THEN BEEP:GOT
O 1250
1260 IFA$="Y"ORA$="y"THEN KEYON:END
1270 IFA$="N"ORA$="n"THENRUN
```

Se dragens øjne følge cursoren rundt på skærmen i den stjerneklare luft

Nu til HARD ROCK

```
0  WIDTH40
1 ' Iron Maiden
2 ' --------------------------------
--
3 ' Made by Garf xxx(Rockin all ove
r)
4 '
5 ' Greetings to all hardrock and
6 ' computer-fans
7 '
8 ' April 1988
9 ' --------------------------------
--
10 COLOR6,1,1:SCREEN2
20 DEFUSR=&H41:A=USR(0)
30 FORI=0TO240
40 READA$
50 VPOKEI,VAL(A$)
60 VPOKE&H2000+I,&H71
70 NEXTI
```

```
80 DATA 231,165,165,164,164,164,164
,164,32,112,216,140,38,243,6,140,1,
3,6,12,24,49,98,196,129,192,96,48,2
4,140,70,35,248,204,102,35,41,46,43
,41
90 DATA 224,160,160,160,160,32,32,1
60,165,165,165,165,231,0,0,0,217,10
8,54,27,13,7,2,0,143,192,96,48,159,
0,0,0,241,3,6,12,248,0,0,0,168,40,4
0,40,56,0,0,0,160,160,160,160,224,9
6,32,0
100 DATA 252,102,51,17,16,16,23,21,
4,14,25,177,225,1,29,181,0,0,16,56,
109,55,24,8,30,50,98,202,154,58,74,
138,115,82,82,82,82,82,82,82,192,96
,48,152,204,230,179,153
110 DATA 31,16,23,20,20,23,16,151,2
39,38,227,1,1,225,33,225,199,101,53
,29,77,113,89,77,20,20,20,20,28,0,0
,0,229,69,5,5,7,3,1,0,24,55,109,56,
16,0,0,0,74,42,154,194,126,0,0,0
120 DATA 82,82,82,82,115,0,0,0,140,
153,243,6,252,0,0,0,212,148,23,8,7,
0,0,0,1,1,225,33,225,0,0,0,69,69,69
,69,199,3,1,0,0,0,0,0,0,0,0,0,0,0
130 VPOKE&H1800,0:VPOKE&H1801,1:VPO
KE&H1802,2:VPOKE&H1803,3:VPOKE&H180
4,4:VPOKE&H1805,5
140 VPOKE&H1820,6:VPOKE&H1821,7:VPO
KE&H1822,8:VPOKE&H1823,9:VPOKE&H182
4,10:VPOKE&H1825,11
150 VPOKE&H1840,12:VPOKE&H1841,13:V
POKE&H1842,14:VPOKE&H1843,15:VPOKE&
H1844,16:VPOKE&H1845,17
160 VPOKE&H1846,18:VPOKE&H1847,19:V
POKE&H1848,20:VPOKE&H1860,21:VPOKE&
H1861,22:VPOKE&H1862,23
170 VPOKE&H1863,24:VPOKE&H1864,25:V
POKE&H1865,26:VPOKE&H1866,27:VPOKE&
H1867,28:VPOKE&H1868,29
180 FORA=&H1806TO&H181F:VPOKEA,30:N
EXT
190 FORI=2047TO2175
200 READA$
210 VPOKEI,VAL(A$)
220 VPOKE&H2000+I,&HF1
230 NEXTI
240 DATA0,0,0,0,1,3,3,7,0,7,63,255,
255,239,245,245,0,224,254,255,255,1
91,183,215,0,0,0,0,128,192,224,224,
14,13,0,25,88,120,88,88,243,247,0,2
55,252,59,67,2,175,31,67,92,63,128,
132,192,176,112,112,0,48,48,32,48
250 DATA28,62,15,3,2,1,6,7,5,62,253
,239,184,69,213,192,224,49,95,231,3
,105,107,3,120,232,232,176,96,64,12
8,0,7,3,1,0,0,0,0,0,192,206,199,85,
81,46,30,14,35,199,198,38,172,24,24
0,224,0,0,0,0,0,0,0,0
260 VPOKE&H190D,1:VPOKE&H190E,2:VPO
KE&H190F,3:VPOKE&H1910,4:VPOKE&H192
D,5:VPOKE&H192E,6:VPOKE&H192F,7:VPO
KE&H1930,8:VPOKE&H194D,9:VPOKE&H194
E,10:VPOKE&H194F,11:VPOKE&H1950,12:
VPOKE&H196D,13:VPOKE&H196E,14:VPOKE
&H196F,15
270 FORI=&H1900TO&H190C:VPOKEI,0:NE
XT
280 OPEN"grp:"AS#1
290 FORI=1TO2
300 PRESET(15+I,150):PRINT#1,"GARF
XXX PRESENTS IRON MAIDEN.":PRESET(8
0+I,160):PRINT#1,"The Trooper":NEXT
I
310 DEFUSR=&H44:A=USR(0)
320 PLAY"O4S1M3400T200D4"
330 PLAY"D8D8D4D8D8D4D8D8D4D8D8D4D8
D8D4D8D8"
340 PLAY"C4F4C4D2D8D8D4D8D8D4D8D8D4
D8D8D4D8D8D4D8D8"
350 PLAY"C4F4C4O3B-2B-8B-8B-4B-8B-8
B-4B-8B-8B-4B-8B-8B-4B-8B-8B-4B-8B-
8"
360 PLAY"O4C4F4C4D2D8D8D4D8D8D4D8D8
D4D8D8D4D8D8D4D8D8"
370 PLAY"C4F4C4D2":GOTO330
```

Og Nu til slut SCARBOROUGH FAIR

```
0  ' Mattias Gustafsson
1 CLS:PRINT"SCARBOROUGH FAIR":PRINT
"----------------"
5 PLAY"S10M10000","S10M10000","S10M
10000"
10 PLAY"O3A2AO4E2EO3B.O4C8O3BA2.R4O
4EGA2G","O3E2EA2CGGGA2.R4CBA2B","O3
A2AA2ADDDE2.R4CED2E"
20 PLAY"EF#DE2.R2AA2AG2EEDC","CA2B2
.R2BC2DE2CCBA","ADF#E2.R2EA2DC2CCGA
"
30 PLAY"B8A8G2RA2ED2CBAG8A8A.","E2.
RE2.F2GGF2E.","B2.RA2.B-2CD2DA2."
40 PRINT"Isn't he marvellous, Garf
xxx-king of   music? If you think s
o then press space-bar"
50 A$=INKEY$:IFA$=""THEN50
60 IF A$=" " THEN 10ELSE PRINT "I w
ill never boast again!"
70 END
```